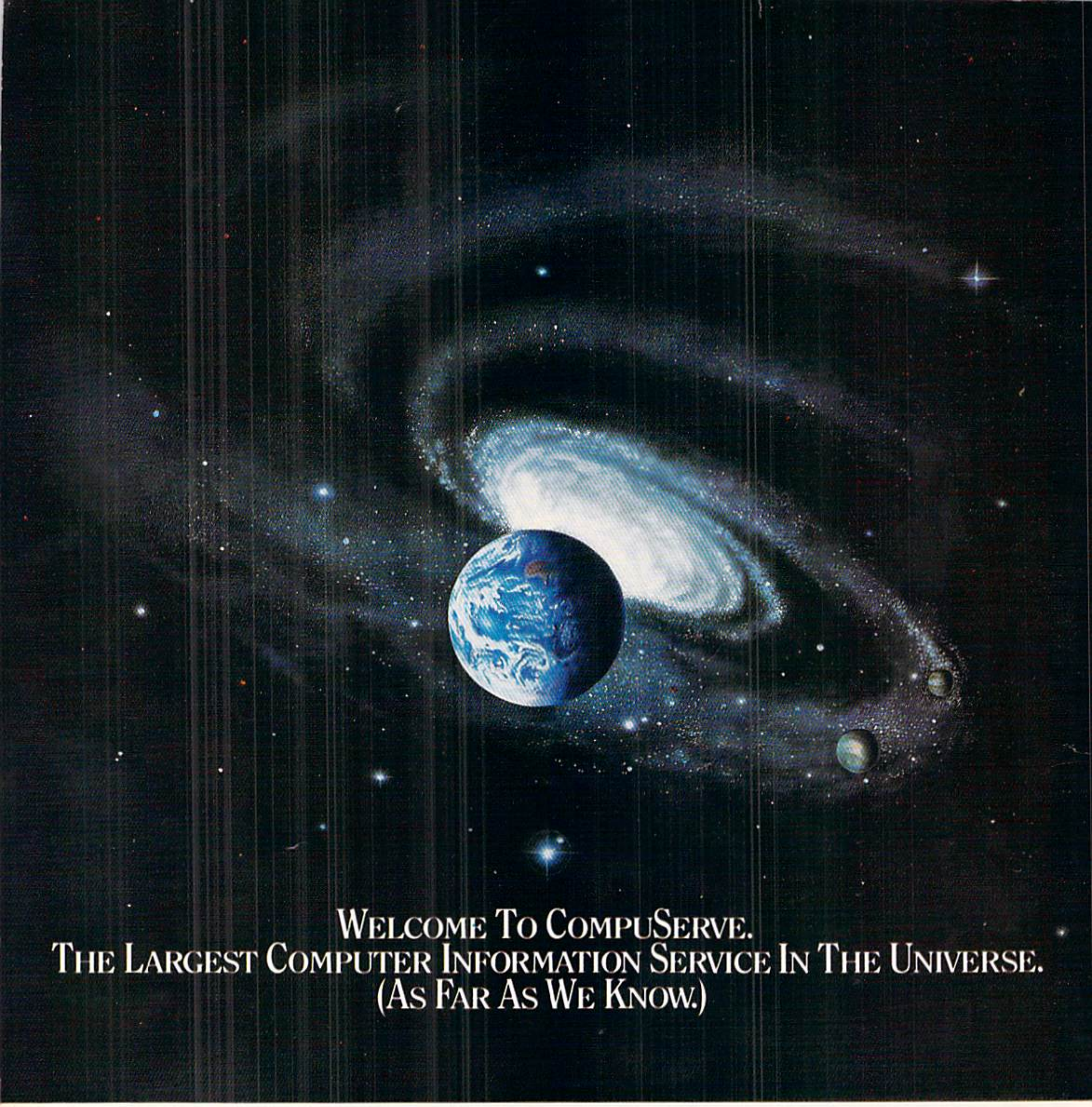
COMMODORE USERS

Ahoy!

JULY 1988





More people use CompuServe than any other online computer information service on Earth.

Over 380,000 members access hundreds of information, communications and entertainment services online.

Thousands with similar interests and special expertise converge regularly and exchange ideas on an ever-expanding list of special-interest Forums.

They have access to a combination of more financial information, a greater variety of shopping services, and deeper research resources than any other online computer service. Anywhere.

Of course, it's conceivable that there's a service like ours somewhere that could give you more for your money. But you may have to travel a few light-years to find it.

Instead, all you have to do is visit your nearest computer dealer today. To order CompuServe direct, or for more information, write: CompuServe, 5000 Arlington Centre Blvd., Columbus, Ohio 43220. Or call 800 848-8199. In Ohio and Canada, call 614 457-0802.

Reader Service No. 145

CompuServe®

AN H&R BLOCK COMPANY

Ahoy!

CONTENTS

DEPARTMENTS

FEATURES

PROGRAMS

Cover art by James Regan; photography by Michael R. Davila

President
Michael Schneider

Publisher
David Allikas

Executive Editor
Michael R. Davila

Art and Production Director
Laura Palmeri

Senior Editor
Cleveland M. Blakemore

Technical Editors
David Barron
Bob Lloret

Consulting Editors
Morton Kevelson
Tim Moriarty
Dale Rupert

Entertainment Editor
Arnie Katz

Art Production
Christopher W. Carter

Circulation Director
W. Charles Squires

Advertising Director
Lynne Dominick

Director of Promotion
Mark Kammerer

Promotion Art Director
Stacy Weinberg

Controller
Dan Tunick

Advertising Representative
JE Publishers' Representative
6855 Santa Monica Blvd., Suite 200
Los Angeles, CA 90038
(213) 467-2266
Boston (617) 437-7628
Dallas (214) 660-2253
New York (212) 724-7767
Chicago (312) 445-2489
Denver (303) 595-4331
San Francisco (415) 864-3252

ISSUE NO. 55 JULY 1988

Ahoy! (ISSN #8750-4383) is published monthly by Ion International Inc., 45 W. 34th St., Suite 500, New York, NY 10001. Subscription rate: 12 issues for $23.00, 24 issues for $44.00 (Canada $30.00 and $55.00 respectively). Second class postage paid at New York, NY 10001 and additional mailing offices.

Postmaster, send address changes to *Ahoy!*, 45 W. 34th Street, Suite 500, New York, NY 10001. Direct all address changes or matters concerning your subscription to *Ahoy!*, P.O. Box #341, Mt. Morris, IL 61054 (phone: 815-734-4151). All editorial inquiries and products for review should be sent to *Ahoy!*, 45 W. 34th St., Suite 500, New York, NY 10001.

VIEW FROM THE BRIDGE

We have several longer, possibly more significant features in this month's issue that we *could* begin this column by talking about—but to us, this month's biggest news is the return of Buck Childress with his first full-length program in six months. We join Buck in thanking the many readers who've written to ask about his progress. Happily, the eye ailment that sidelined our most prolific programmer is all but licked, and Buck's output should rise steadily in the months to come. (As for Buck's program, *VariScan*, it safeguards C-64 and C-128 programmers from reusing variable names and crashing works in progress. Turn to page 30.)

After good news like that, we hope you won't find our description of the rest of the July *Ahoy!* too anticlimactic:

• As Dale Rupert points out, past *Rupert Reports* have involved connecting your computer to photo-cells, LED's, potentiometers, and relays. This month Dale helps you make the *Thermal Connection,* and turn your 64 or 128 into a digital thermometer. What's next: a Commodore-driven vacuum cleaner? Satellite dish? Roller coaster? Only time will tell. (Turn to page 32.)

• *Window Dressing* means curtains for those drab displays called up by the C-128's WINDOW command. Richard Curcio's quartet of routines let you print attractive boxes and headings, manipulate text, and more. (Turn to page 37.)

• Adventure game fans registered one complaint about *Vault of Terror* (Oct. '86) and *Crypt of Fear* (Feb. '88)—their fingers were trembling so hard that they couldn't type the commands. So Cleve Blakemore designed *Tomb of Horror,* the last entry in his 3-D trilogy, to work entirely via pulldown menus. If you find yourself shaking so much that you can't wiggle a joystick, you're on your own! (Turn to page 48.)

• Mindful that some readers don't like typing in listings of 8+ pages in length, Cleve offers two games that can easily be entered in a single sitting. *Guerilla* lets you go bananas gunning down terrorists who shoot at you from the ruins of bombed-out buildings. (Turn to page 16.) And *Lunar Buggy* dispatches you on an equally patriotic mission—to recapture the stolen artifact of the Apollo space program and drive it across treacherous terrain to safety. (Turn to page 15.)

• Once you've played all three of the above and asked the inevitable question—"How does Cleve Blakemore do it?"—refer to *Programming Your Own Text Games,* in which Cleve continues to reveal his secrets. (Turn to page 13.)

• Once again, Arnie Katz and the undersigned have been chosen (along with Betsy Staples of *Atari Explorer*) to select the programs for inclusion in this year's CES Software Showcase. We're proud to feature game reviews by Arnie and his partners, Bill Kunkel and Joyce Worley, in every issue. This month's *Entertainment Software Section* covers *Stealth Mission, Speed Buggy, Wooden Ships & Iron Men,* and *Plasmatron.* (Turn to page 20.)

• If you've been putting off looking into our *COMAL Column,* you're almost out of time. Richard Herring presents the penultimate installment in this issue. (Turn to page 45.)

In case anyone is still confused by our division into *Ahoy!* and *Ahoy!'s AmigaUser,* remember—the next *Ahoy!* for the C-64 and C-128 will be the September issue, on sale August 2. On the stands next month will be the second issue of *Ahoy!'s AmigaUser*—cover-dated August, and on sale July 5. And remember, if you're upgrading to the Amiga and want to switch your *Ahoy!* subscription to *Ahoy!'s AmigaUser,* you can—but please call 815-734-4151 or write *Ahoy!*, P.O. Box #341, Mt. Morris, IL 61054. Do not call us at our New York number with subscription problems—we can't help you from here. If you have any *non*-subscription questions, we'd love to help you. Call or write anytime.

—David Allikas

Free Spirit Software Inc.

UTILITIES

THE SUPER CHIPS
Custom Operating System for the C128

Three 16K ROM chips that add several powerful features to Basic 7.0 including FIND, CHANGE...THIS...TO...THAT, TYPE, UNNEW, COMBINE, MERGE, START, FILE, EDITOR and more! Simultaneous split screen directories of devices 8 & 9. Compatible with 1541/1571/1581 and virtually all software and peripherals. Only **$49.95!**

Super Chips, Custom Operating System for the C128D - Two 32K ROM chips - Only **$49.95!**

Super Chip, Custom Operating System for the C64 - One 16K ROM chip - Only **$29.95!**

Super Chip, Custom Operating System for the 64 mode of the C128 - Only **$29.95!**

SUPER AIDE

All-purpose utility program for the C64 provides:

- Bi-directional scrolling
- Auto Line Deletion
- Trace function
- Disassembler
- Lo-Res Screen Dump
- Number conversion (10, hex, binary)
- Append files
- Format — short new/complete new
- Menu-driven
- Auto Line Numbering
- Renumber
- ML Monitor
- List all variables to screen
- Hi-Res Screen Dump
- Restore newed Basic program
- Change Device number
- Packed Line Editor
- Determine file load address
- Change THIS TO THAT — search for all instances of specified string and replace with second specified string

• And much, much more!

Super Aide, the complete programmer's tool kit. Only **$29.95!**

". . . excellent, efficient program that can help you save both money and downtime."

Compute!'s Gazette Dec., 1987

1541/1571 DRIVE ALIGNMENT

1541/1571 Drive Alignment reports the alignment condition of the disk drive as you perform adjustments. On screen help is available while the program is running. Includes features for speed adjustment. Complete instruction manual on aligning both 1541 and 1571 drives. Even includes instructions on how to load alignment program when nothing else will load! Works on the C64, SX64, C128 in either 64 or 128 mode, 1541, 1571 in either 1541 or 1571 mode! Autoboots to all modes. Second drive fully supported. Program disk, calibration disk and instruction manual only **$34.95!**

SUPER 81 UTILITIES

Super 81 Utilities is a complete utilities package for the 1581 disk drive and C128 computer. Among the many Super 81 Utilities features are:

- Copy whole disks from 1541 or 1571 format to 1581 partitions.
- Copy 1541 or 1571 files to 1581 disks
- Backup 1581 disks or files with 1 or 2 1581's
- Supplied on both 3½" and 5¼" diskettes so that it will load on either the 1571 or 1581 drive.
- Perform numerous DOS functions such as rename a disk, rename a file, scratch or unscratch files, lock or unlock files, create auto-boot and much more!

Super 81 Utilities uses an option window to display all choices available at any given time. A full featured disk utilities system for the 1581 for only

Super 81 Utilities is now available for the C64!

$39.95!

RAMDOS Lightning Fast RAM-DISK

RAMDOS is a complete RAM based "Disk" Operating System for the Commodore 1700 and 1750 RAM expansion modules which turns all or part of the expansion memory into a lightning fast RAM-DISK. RAMDOS behaves similar to a much faster 1541 or 1571 floppy disk except that the data is held in expansion RAM and not on disk. Under RAMDOS, a 50K program can be loaded in ½ second. Programs and files can be transferred to and from disk with a single command. RAMDOS is available for only **$39.95!**

GAMES

EYE OF THE INCA

Four text adventures on one disk for the C64 and Apple II series computers. Eye of the Inca, Shipwrecked, Son of Ali Baba and Perils of Darkest Africa. Four perilous adventures for only **$19.95!**

REVENGE OF THE MOON GODDESS

Four text adventures on one disk for the C64 and Apple II series computers. Revenge of the Moon Goddess, Frankenstein's Legacy, Night of the Walking Dead and The Sea Phantom. Four terrifying adventures for only **$19.95!**

SEX VIXENS FROM SPACE

Three text adventures for the C64 and Apple II series for MATURE ADULTS ONLY. Sex Vixens from Space, Bite of the Sorority Vampires and Hatchet Honeymoon. Three sizzling adult adventures for only **$29.95!**



SUPER BIKE

Action-packed, fun-filled motor cycle arcade game for the C64. Race the clock in Motocross, Enduro, Supercross or Trials. Fly through the air on spectacular jumps. Bounce over woop-de-doos.

Avoid logs, trees, water holes, brick walls, other bikers, etc. as you vie for the gold cup.

Thrilling **Super Bike** action for only **$14.95!**

GALACTIC FRONTIER

Exciting space exploration game for the C64. Search for life forms among the 200 billion stars in our galaxy. Scientifically accurate. Awesome graphics! For the serious student of astronomy or the casual explorer who wants to boldly go where no man has gone before.

Only **$29.95!**

Order with check, money order, VISA, MasterCard, COD. Free shipping & handling on US, Canadian, APO,FPO orders. COD & Foreign orders add $4.00

Order From: **Free Spirit Software, Inc.**
905 W. Hillgrove, Suite 6
LaGrange, IL 60525
(312) 352-7323
1-800-552-6777
For Technical Assistance call: **(312)352-7335**

In England contact Financial Systems Software 0905-611-463

SCUTTLEBUTT

GAMES FROM COSMI, DATASOFT, EA, MICROILLUSIONS • C-128 BBS PROGRAM • DISK DRIVE ACCESSORY • CP/M NEWS • MUSIC SOFTWARE • 1581 UTILITIES FOR C-64 • ONLINE EXPORTER SERVICES • SEX EDUCATION • DESKTOP PROGRAM • TEACHERS' CONTEST • AI TYPING TUTOR

EXPORTER SERVICES

A new online system offered through CompuServe allows US firms to identify export opportunities and speed up the processing of export licenses. Under the terms of a contract with the US Dept. of Commerce (DOC), exporters can submit their license applications to DOC electronically. A DOC licensing officer will process the application, send a response to the exporter electronically, and follow up with a hard copy of the validated license.

Also new to CompuServe is a database providing detailed information, updated daily, on worldwide trade opportunities. Exporters can search the database by country, SIC code, posting date, and type of procurement.

CompuServe, 614-457-8600 (see address list, page 12).

ILLINOIS SHOW

The third annual Chicagoland Commodore Computer Fest is scheduled for August 28 at the Exposition Center at the Kane County Fairgrounds in St. Charles, IL. National speakers and 64, 128, and Amiga vendors will be featured. Admission is $5.00.

Computerfest, 312-897-5788 (see address list, page 12).

LET ME TELL YOU 'BOUT...

Why embarrass yourself by your lack of knowledge? *Birds 'N Bees* lets your kids learn the facts of life interactively. The program, designed by a team of psychologists and consultants, includes treatment of reproduction and child development, plus a special section on protecting children from strangers. The program is designed to let children of any age and either gender receive information specific to their needs. Price is $49.95 for the C-64, $59.95 for the Amiga.



Sex education for kids of all ages.
READER SERVICE NO. 174

IntraCorp, Inc., 305-252-9040 (see address list, page 12).

SUPER 81 FOR THE 64

A C-64 adaptation of *Super 81 Utilities* joins the C-128 version introduced earlier by Free Spirit. *Super 81/64* will copy whole disks or files from 1541/71 drives to the 1581. The user can back up disks or files with one or two 1541's, 1571's, or 1581's, or any combination thereof. Also included are a full-featured sector editor and utilities for partitioning, scratch and unscratch, lock and unlock, rename, format, and direct DOS commands.

The program is supplied on both 5¼" and 3½" disks, and boots on either device 8 or 9. Price is $39.95; shipping is free.

Free Spirit Software, Inc., 312-352-7323 (see address list, page 12).

PAPERCLIP PUBLISHER

PaperClip Publisher ($49.95) offers all the text formatting, layout, and design capabilities needed for creating multiple-column page layouts for newsletters, flyers, price lists, menus, greeting cards, and the like. Documents up to 50 pages in length are supported. Other special features are the ability to temporarily hold and work on text or graphics outside a given document, and a "refreshing" feature that automatically updates any altered section.

Electronic Arts, 415-571-7171 (see address list, page 12).

MUSIC SOFTWARE

Low-cost music programs for the pro or semi-pro:

Passport Sequence Editor ($14.95) allows step-editing of the MIDI 4+ and 8+ sequence files.

Studio One Editor ($14.95) works with Syntech's *Studio One* program.

DX21/27/100 Librarian ($14.95) stores banks and individual voices from Yamaha's 4-operator FM synthesizers.

Generic Librarian ($19.95), a 32K system-exclusive recorder, works with any instrument capable of bulk Sys-Ex dumps.

SoundWare (see address list, page 12).

CP/M STARTER SET

The PDS *CP/M Starter Set* ($29.95) is comprised of four disks of utilities and applications for the C-128, plus printed documentation explaining booting up, transient and resident commands, and creating and dissolving library files.

Public Domain Solutions, 813-378-2394 (see address list, page 12).

DRIVE ENHANCEMENT

Microteq's *Drive Box* ($29.95) permits hardware configuration of the 1541, the 1571, and the C-128D's built-in drive. The box allows for setting the

CHIP LEVEL DESIGNS PRESENTS

128 mode & CP/M version now available!

- RapiDOS -

THE SUPER-FAST PARALLEL DISK OPERATING SYSTEM FOR THE COMMODORE 64 AND 1541 DISK DRIVE!

- **All** disk access is handled at super-fast parallel speed! (LOAD, SAVE, directory, SEQ & REL files, scratch, validate, format).
- Designed to support multiple drive systems.
- Parallel Centronics printer support with file spooling capability.
- Many useful, timesaving features (DOS Wedge, screen dump, resident monitor).
- No loss of compatibility.
- Far too many features to list in this ad . . . and perhaps in this magazine! (Call or write to get all the details!)

. . . and if you want the ultimate, get RapiDOS Professional!

- Gives even faster disk access!
- Uses 8k RAM track buffering and hardware GCR conversion!
- Provides 40 track extension (749 blocks free!)
- Adds 20 new disk commands (i.e., lock files, change disk name).

Here's what people are saying about RapiDOS:

Mike J. Henry (Basement Boys Software) - *"It's amazing how incredibly fast it is, I'm impressed!"*
Mitch S. (Eaglesoft Inc.) - *"Very fast, very reliable, and very compatible. I love it!"*
J.F. Jones (ADP) - *"Superbases' speed is increased greatly, and it's now a dream to use!"*

Function	Normal DOS	RapiDOS	RapiDOS Pro	Your System
Load 202 blocks	128 sec.	15 sec.	3 sec.	
Save 202 blocks	196 sec.	98 sec.	8 sec.	
Format 35 tracks	90 sec.	24 sec.	18 sec.	

Compare these speeds with your current system and see why RapiDOS puts the C-64 into a different league!

RapiDOS requires a socketed kernal ROM U4, and is available in versions for the 64c, 128 in 64 mode, and 1541c (please specify when ordering). RapiDOS is easily upgradeable to the Professional Version. RapiDOS Professional drive controller is **(c) '87 mts data GbR,** the creators of the best European parallel systems.

At these affordable prices no C-64 owner should be without RapiDOS!

RapiDOS $49.95 **RapiDOS Professional $99.95**

MASS DUPLICATOR 1541

- For the C64/128 with a single 1541 disk drive.
- 15 second, 4 pass backup for standard disks!
- 25 second full GCR Super Nibbler, the most powerful yet!
- 9 second disk format!
- Fast loader!
- Quick installation. **$32.95**
- Can be upgraded to RapiDOS

MASS DUPLICATOR MSD

- A must for any MSD SD-2 owner.
- 15 second standard disk backup!
- 18 second full GCR Quick Nibbler!
- 9 second disk format! **$25.95**

MSD AUTO COPY ROM

- Adds new Fast Backup commands!
- Turns the MSD SD-2 into a dedicated copying drive (no computer needed). **$29.95**

C-64 BURST-ROM

- Gives the 64 'Burst Mode' when used with a 1571 or 1581 disk drive!
- Loads 100 blocks in 6 seconds on a 1571, 4 seconds on a 1581!
- Fast directory, SEQ, and REL file access!
- Built in DOS wedge!
- Simple installation. (Kernal ROM U4 must be socketed) **$32.95**

C-128 BURST-ROM

- Lets the 128 run at 'Burst' speed when in 64 mode!
- Provides the same features as the C-64 Burst-ROM! **$38.95**

STILL TO COME

- MSD SD-1 Fast Copier!
- 1571 Mass Duplicator!
- 1581 Utility Pack!

TURBO 64

- Speed-up cartridge for the C-64, just plug it in!
- Adjust the clock rate from 100 khz (1/10 normal) to 4 mhz (4x normal)!
- Uses 8/16 bit 65816 microprocessor (same as the Apple][gs... but twice as fast)!
- Spread sheets, BASIC, flight sims, graphics, and now GEOS... all are accelerated! **$189.95**

CLD

CHIP LEVEL DESIGNS

Cash, Check, Money Order, M.C. or Visa

$3.00 shipping on all orders

C.O.D.'s add $3.00

P.O. BOX 603
ASTORIA, OR 97103-0603

(503) 861-1622

Dealer, Distributor, & Group Pricing Available

device number from 8-11, as well as for writing to the reverse side of the disk without cutting an extra notch. Six solder connections are required in order to install the 3⅛ x 2 x 1" unit; or, Microteq will install it in your drive for $10 plus shipping (you must insure your drive).

Drive Box is available through Software Support and Free Spirit.

Microteq Systems, 701-232-4033 (see address list, page 12).

KETEK SUPPORT

Computer accessories from Ketek, makers of the Command Center line of system enclosures for the 64, 64C, and 128:

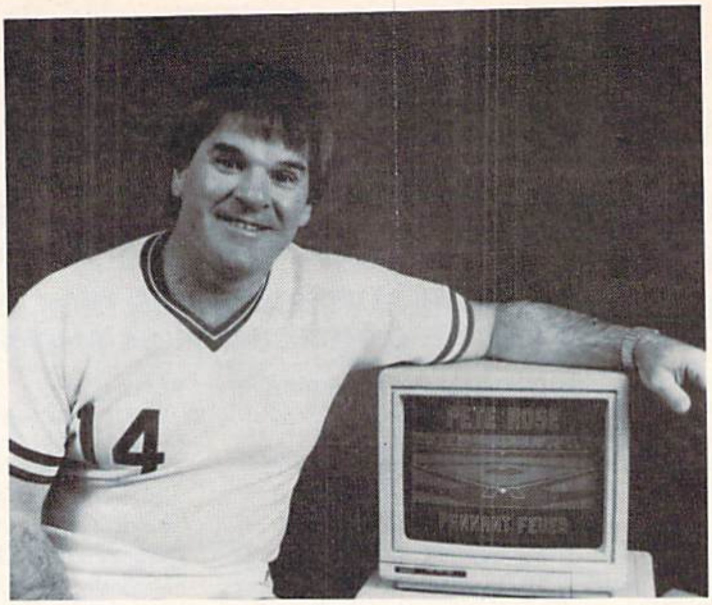
The Sound Trap ($49.95) encloses almost any 80 column printer with sound-absorbing foam and wood to reduce noise by up to 90%. Included are an acrylic lid and a slide-out shelf for catching printout.

The Remote Power Controller ($89.00) provides outlets for up to five devices, each offering 3-way protection against surges and spikes. The base unit can mount behind or under your desk, while the 3⅜ x 5¾ x 1¼" controller is kept close at hand.

Data Director ($39.95) makes it possible for two 6-pin serial devices to share one 64, or for two 64's to share one serial device. The unit works with all Commodore-compatible drives, printers, and interfaces.

The Tilt/Swivel Monitor Stand ($14.95) lets you turn your monitor to the desired angle, with cushioned pads to hold the monitor securely in place.

Ketek, 319-338-7123 (see address list, page 12).



Pete Rose Pennant Fever ***incorporates the strategies of the Reds star. You can hit, run, field, and presumably throw body blocks at opposing catchers.***
READER SERVICE NO. 175

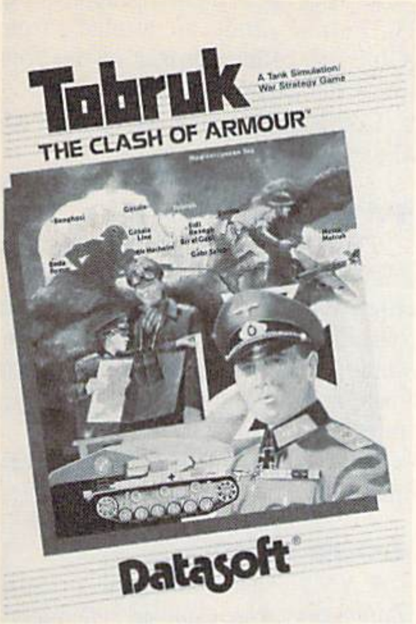
GAMES

Scheduled for C-64 release in November, *Pete Rose Pennant Fever* lets you guide an expansion team through a 10-season, 24-team race for the pennant. Rather than looking down on the diamond from overhead, you experience the action from the perspective of one of Charlie Hustle's teammates. You pitch, hit, run, field, throw, and steal as you would on a real diamond. Managerial strategies actually employed by Rose are available. In addition, you can act as General Manager, drafting and acquiring computerized players while maintaining control of the club's finances and player salaries. Is there a collusion option? Wait and see.

Activision/Gamestar, 415-960-0410 (see address list, page 12).

Two for the 64 from EA:

Scheduled for late summer release, *Wasteland* ($49.95) challenges players to survive in the post nuclear year of 2087. As you and your band of Desert Rangers roam the southwestern United States, trying to help other survivors rebuild and live in peace, you'll en-



Tank's-eye view of the WWII clash.
READER SERVICE NO. 176

counter a cast of hundreds, including mutant villains who endanger the population. At certain points in the game, you'll be directed to the included Paragraphs book which enhances the detail of the game and provides descriptive text. After the game is over, you may return to explore locations you were not able to the first time around.

Patton vs Rommel ($29.95), designed by Chris Crawford, lets the player become either General Blood and Guts or the Desert Fox on D-Day. Each commander has numerous infantry and armor divisions at his disposal, plus a historically and geographically accurate map of 150,000 locations. 10 different kinds of tactical orders are possible, and each division can have up to 32 different orders at a time. To give orders, the players just point at a division and an objective–the army does the rest. As the strategies come to life, players can watch the battles take place, with realistic sound effects. For the C-64.

Electronic Arts, 415-571-7171 (see address list, page 12).

Rommel stars as well in *Tobruk* ($19.95), in which you command 10 divisions of the Afrika Korps in an attempt to capture the key Libyan port city. Action can be controlled from tactical maps, or directly in land and sea battles. Players view the battle from the turret of a desert tank, with access to machine gun controls and mines. Air tactics include ground attacks, air-to-air skirmishes, and long-range bombing. The C-64 simulation is designed by Datasoft and distributed by Electronic Arts.

Electronic Arts/Datasoft, 415-571-7171 (see address list, page 12).

Under Fire! ($34.95) consists of nine WWII scenarios involving the US, Germany, and the Soviet Union, plus a construction set enabling the C-64 gamer to create more.

Avalon Hill, 301-254-9200 (see address list, page 12).

Three C-64 racing games–*Richard Petty's Talladega, Shirley Muldowney's Top Fuel Challenge,* and *Grand Prix Motor Mania*–have been combined into Cosmi's *Motor Racing Trilogy* ($24.95).

Cosmi, 714-240-8985 (see address list, page 12).

Black Jack Academy will teach you to play the game, or help you brush up your skills. Online help explains all play options and coaches you along. Special ease-of-use features include multiple play speeds, card counting option, player card total option, money management guide, and preset multiple table rules from Las Vegas, Reno, and Atlantic City. For the 64 or Amiga; $39.95.

MicroIllusions, 818-360-3715 (see address list, page 12).

Bridge Baron II ($39.95), an improved version, plays the complete game of bridge, allowing you to bid and play more than a billion different deals with your C-64 as both your partner and your opponents. New features include two-person mode, rubber bridge scoring, automatic play mode, Baron's recommended bids and plays, the ability to claim or concede tricks, simplified card play, weak 2-bids, and the option to save deals to disk.

Great Game Products, 800-GAMES-4-U (see address list, page 12).

MORE CP/M

Poseidon has published the spring '88 addendum to its catalog of CP/M software, priced at $1.75 plus a 45¢ SASE (if paying by check, it must be made out to Ralph Lees). New software prices are $16 for the first disk, $12 for the second, and $7 each for all subsequent ones.

Poseidon Electronics, 212-777-9515 (see address list below).

BBS FOR 128

Dragonfire BBS 128 ($100) is designed to allow a novice to get a bulletin board up and running within 20 minutes. It supports 300, 1200, and 2400 baud modems, up to 100 public and 40 private message bases, full E-mail system, remote SYSOP capabilities, auto maintenance mode, security option, self-maintaining system files, and more. Versions for the 64 and Amiga are forthcoming, as are terminal programs for the 64 and 128.

YodaHead Software, 609-596-1772 (see address list below).

SWEEPSTAKES

Computer Learning Month has launched a School Certification Program and Sweepstakes, with a first prize of a computer and software and 30 second prizes. To become eligible, teachers in grades K-12 must use three software programs they have not tried before between August 1 and October 31. Further details are available from CLM, an official project of the Software Publishers Association.

Computer Learning Month (see address list below).

PAINT PROGRAM

The *Masterpiece* graphics package ($29.95) lets the C-64 artist move, copy, scale, rotate, twist, and fold the onscreen artwork. Other features include picture compression for saving memory, and the ability to load pictures from BASIC.

Scorpion, 201-663-0202 (see address list below).

GOOD NEWS

P.A.V.Y. has lowered the price of *Landmark, The Computer Reference Bible* from $164.95 to $129.95. If you're reading these words before June 1 (as only subscribers are likely to be), you may still be able to get in on P.A.V.Y.'s limited time special price of $119.95

Continued on page 81

Companies Mentioned in *Scuttlebutt*

Contact manufacturers directly for additional information.

Activision/Gamestar
2350 Bayshore Parkway
Mountain View, CA 94043
Phone: 415-960-0410

Avalon Hill
4517 Harford Road
Baltimore, MD 21214
Phone: 301-254-9200

CompuServe
5000 Arlington Centre Blvd./P.O. Box 20212
Columbus, OH 43220
Phone: 614-457-8600

Computer Learning Month
P.O. Box 60007
Palo Alto, CA 94306-6007

Computerfest
P.O. Box 28
North Aurora, IL 60542
Phone: 312-897-5788

Cosmi
415 North Figueroa Street
Wilmington, CA 90744
Phone: 714-240-8985

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
Phone: 415-571-7171

Free Spirit Software
905 W. Hillgrove, Suite 6
LaGrange, IL 60525
Phone: 312-352-7323

Great Game Products
8804 Chalon Drive
Bethesda, MD 20817
Phone: 800-GAMES-4-U

IntraCorp, Inc.
14160 S.W. 139th Court
Miami, FL 33186
Phone: 305-252-9040

Ketek
P.O. Box 203
Oakdale, IA 52319
Phone: 319-338-7123

MicroIllusions
17408 Chatsworth Street
Granada Hills, CA 91344
Phone: 818-360-3715

MicroProse
180 Lakefront Drive
Hunt Valley, MD 21030
Phone: 301-771-1151

Microteq Systems
1430 9th Avenue South
Fargo, ND 58103
Phone: 701-232-4033

P.A.V.Y. Software
P.O. Box 1584
Ballwin, MO 63022
Phone: 314-527-4505

Poseidon Electronics
103 Waverly Place
New York, NY 10011
Phone: 212-777-9515

Public Domain Solutions
P.O. Box 832
Tallevast, FL 34270
Phone: 813-378-2394

Scorpion
19 Harbor Drive
Lake Hopatcong, NJ 07849
Phone: 201-663-0202

The Software Toolworks
One Toolworks Plaza
13557 Ventura Blvd.
Sherman Oaks, CA 91423
Phone: 818-907-6789

SoundWare
P.O. Box 1913
Nederland, TX 77627

YodaHead Software
P.O. Box 177
Marlton, NJ 08053
Phone: 609-596-1772

PROGRAMMING YOUR OWN TEXT GAMES

Part II of III: Think Modules!

By Cleveland M. Blakemore

Ingenuity lies not in further complicating definitions and concepts, but in reducing the number of contradictions and terms to the fewest possible true sets. Any given system design must begin with the sum of its parts, and then define these parts.

This is the essential theory behind "black box" system design, or modular programming. We begin by describing what we want the system to do, then breaking this idea down into the necessary steps to accomplish it. Whether you are pursuing a personal goal or creating a computer program, you will find that this approach is always the most effective.

Last month we published a short text adventure called *Lost Dutchman's Mine.* In this installment we will examine this program in detail and demonstrate how a series of modules (or subroutines) can be tied together into one cohesive system that is greater than the sum of its parts. None of the routines by themselves would constitute a game—but working in tandem, they create an interactive adventure.

If we wanted to describe the player's purpose in *Lost Dutchman's Mine* in one sentence, we could say:

"To move around within a network of locations, collect various objects, return them to a specific location, and win the game."

This is the simplest way to define the program's operation. In last month's column, I advised designing your adventure on three separate pieces of paper before touching the keyboard. Page one of my scenario for *Lost Dutchman's Mine* read like this:

"The object of the game is to wander around within a series of caves, collecting up three treasures as you go. When you have all three objects, you return them to a specific base location, drop them, and win the game."

On page two I listed the obstacles I wanted to hamper the player's efforts:

"1. A timber wolf guards the entrance to a burial mound where one of the treasures is buried. The player must find the gun and bullets to kill the wolf.

"2. A secret word is required to pass from the base location to the caverns. (Kind of like PLUGH in *Colossal Caves.*) The secret word is written inside a matchbook cover.

"3. A whistling spirit guards the entrance to the caverns. The player must blow on an empty 7-UP bottle to scare the spirit away.

"4. The burial mound is submerged under water. The player must find and open a reservoir lock through which the mound room can be drained.

"5. A shovel is needed to dig up the third treasure (Dutchman's nugget).

"6. The player must also light a torch, drop enough equipment to fit through a narrow crack, and escape from the room after he opens the reservoir lock."

You may have noticed that a lot of these ideas are taken from other adventure games, many of them classic puzzles. I wanted them to be familiar enough that the game could be solved without too much trouble. I could just as easily have made the game much harder.

On the third and final page, I drew a map detailing all nine locations in the game, complete with references to the objects that would be found there, and the various puzzles specific to those spots.

Using the modular approach, I broke the program down into seven discrete boxes.

1. Initialize arrays and variables
2. Read in data
3. Get command
4. Find verb in command
5. Find noun in command (if any)
6. GOSUB to appropriate subroutine pertaining to verb
7. Data lines

Modules 3, 4, and 5 could all be considered part of the parser, the routine that gets and analyzes player input. I define them separately here to make it clear what goes on inside the routine.

The first thing I typed in, as always, were lines 1530-1610. These lines contain descriptions of the nine locations in the game. I immediately followed with comments to accompany each, in lines 1630-1670. These comments are specific observations in each location. For example, the gas station has a "metal plaque above a glass shelf."

Then, using the third page of my notes (the map) as a

guide, I typed in the data for an array of nine dimensions, each dimension having six elements. This dimension will be known as D(9,6). The six elements consist of the directions North, South, East, West, Up, and Down. This is a numerical array, and the number in each element indicates the location to which that direction connects. Any non-zero number is a pathway; otherwise there is no exit in that direction. I did not type in directions for the gas station, because it is only accessible by use of the forementioned "magic word."

In the fashion of any sensible hacker, I jumped to lines 120 through 210 and designed a data loader. Once I made sure that the arrays were dimensioned properly in line 120, I read them in a couple of times in lines 190 and 210 just to make sure everything was going smoothly before I continued. This type-n-run approach is the hacking method, and it works. Trust me.

Now for the tough part. I had to figure out all the verbs and nouns that would be needed for the parser's vocabulary. In lines 1800-1850 I tried to give the program a reasonable dictionary of words to work with for player input. An advanced parser would recognize far, far more than this simple collection of words.

Again, I skipped back to the beginning and made sure these strings loaded in correctly before going on.

Lines 1870 through 1900 are detailed descriptions of the objects that can be carried in the player's inventory. Although an object might be referenced by the parser as "diamo," it will be described on the screen as a "glowing diamond." This gives character and atmosphere to the adventure.

Line 1920 is a list of numbers that tells the program where to place objects initially. The first number indicates the location, the second is the object number itself in the inventory. The − 1, −1 at the end of the line signals the end of the data.

Lines 1940-1970 are end of game messages, for when the player is killed or is victorious. If the player is killed, these lines help tell him where he made his mistake.

With this, I completed the data lines, the major stumbling block of any adventure game. Although I made many changes afterwards and altered these lines considerably before finishing, these lines gave me something to work with, the meat of the text adventure. The remaining program portions manipulate the data loaded in from here in many ways, but it is in these lines that the game gets its substance.

Important variables to be defined are CL (Current Location) and the arrays 0(9,8) and I(8). The former is an array that tells what objects are in what rooms, and the latter is the player's inventory array. I(8) refers directly to the eight objects listed in line 1840, the objects in the game which can be carried. Any non-zero number in element I(1), for example, would indicate that the player is carrying the torch. However, a −1 would mean the torch is burning, while a 1 would mean it is out. I also set up meaningful flag names in line 150. These flags indicate the status of certain items.

In lines 340-540, I fleshed out the parser routines. First the directions, then once the program is recognizing them correctly and responding, I introduce the other verbs. It is better to make sure that all the directions are functioning correctly first before continuing, because a flaw that shows up later might take a lot of work to repair. Make certain that the directions in the game correspond identically to your map. It is very common to make errors here. I often get East and West mixed up while typing them in.

Line 340 gets the player's input. If he enters nothing, the program checks again. Line 370 checks for a single character input (either a direction or the letter "i" for inventory) and acts accordingly. Since the six compass directions match up with the six single character commands perfectly, I just reduced any of these terms to a number between 1 and 6, subtracting 7 if necessary. I can then use this number to reference the corresponding element in direction array D(9,6). Once a match is found for the verb, the number is placed in the variable V.

The verb is the important part. We use the verb to jump to the subroutine that designates the action, so we must have a verb. The noun, on the other hand, is not always necessary for many actions, and so our parser only looks for it. If it finds one, it places the number of the noun into the variable N before branching off to our routine. The subroutine can then check this variable to see if the player is referencing the correct object. For example, if the player gives the command to dig, we check the variable N to determine whether or not he wants to dig in any particular spot, as in DIG MOUND. If N=0, we simply print a standardized message that reads "YOU DIG FOR A WHILE BUT YOU DON'T FIND ANYTHING."

After the parser has both these variables, V and N, assigned with values, it drops through to lines 570-580, the branches. These lines will steer the program flow into the correct verb actions that alter program variables and the game environment.

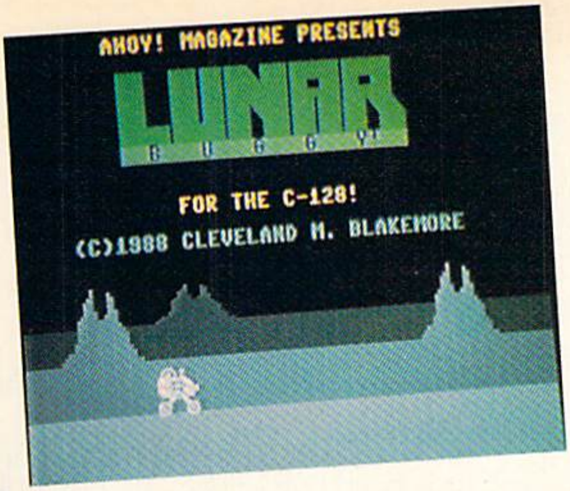
These lines run from 700-1520, accomplishing every possible action that the player is permitted to take in our adventure. If you study them carefully, the variables are self-explanatory and it should be readily apparent that they act on values to change location inventory, specific flags (mound full-empty, wolf dead-alive, etc.), and print messages for the player's benefit.

After program flow returns from these subroutines, the main parser program executes a series of critical checks in lines 600-690. These flags count elapsed time in the current location and check on the player's status relative to certain non-player characters and events. For example, if the player is in room 4 for more than four turns with the spirit, he is going to get it good. The same goes for being in a river full of water too long or beside a hungry wolf.

The check for a win is in line 600. This line checks if a variable called WINGAME has been set yet by the drop subroutine. If the player drops all three treasures in location zero, this flag will be set upon returning from the routine, telling the main program that the player has succeeded.

With as little work as all this, an adventure game was born. We have a full-fledged story, with a plot, characters, and suspense. Next month, we'll go over the specifics of the way the parser functions, and methods we could use to upgrade the routine so that it could recognize complete sentences.

Until then, remember—stop wrestling with the program as a whole and break it down into modules. You cannot move mountains unless you do it one bucket of dirt at a time! □



In the year 2009 A.D. it was discovered that a colony of extraterrestrials had established itself on the dark side of the moon. A Russian installation was attacked and totally obliterated in 2011 when it attempted to approach the alien base to make contact. Following that incident, a United Nations security resolution placed the aliens' location off limits to all international teams until further notice, for fear of antagonizing the otherworldly visitors.

You are a government agent assigned to a secret mission of the highest priority: to recover one of the greatest of all historical artifacts, the Apollo lunar buggy. If you can drive the vehicle over the treacherous lunar plain and past the alien forces, the buggy can be returned to the Smithsonian Museum for posterity. It seems like an impossible gamble, but you know it will be worth it if you can recover this national treasure.

Lunar Buggy is a jump-and-shoot game similar to *Moon Buggy*, an arcade favorite. It requires a joystick in Port 2. Press the fire button to escape from the title screen.

This game for the C-128 features a background that smooth scrolls past at two different speeds to give the illusion of depth. You've probably seen it before in many arcade games. Lunar mountains pass in front of one another as they move, with the smaller and slower moving terrain in the rear of the display.

To jump, you push up on the joystick. Press the button and push either up or right to fire a laser torpedo. You can jump and fire at the same time if necessary.

The alien ships are a mere nuisance at first, but they can become a real threat later on in the game as they begin to move erratically and with greater speed. Destroy them as quickly as possible, but don't let them distract you from jumping over the many craters in the lunar surface.

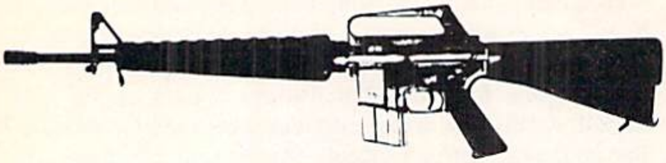
Each time you go 500 miles across the moon towards safety, you'll pass an American outpost. These small bases provide short stops to rest and refuel before moving on.

The alien ships are worth the current base number X 100 points each.

If you manage to go 5000 miles in *Lunar Buggy*, you'll have escaped safely and returned the buggy to American soil, where it belongs. ☐

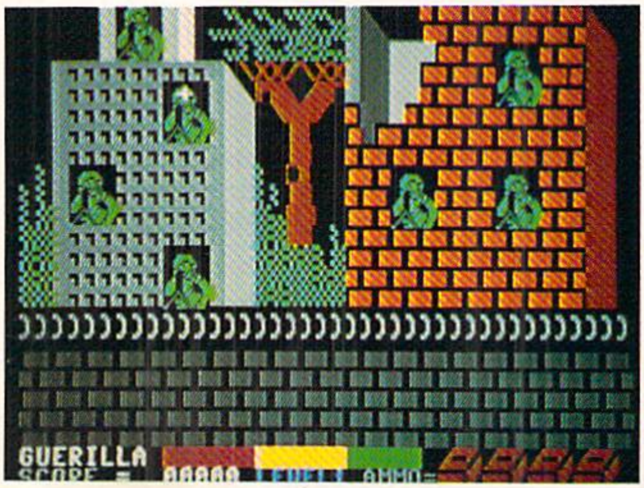
SEE PROGRAM LISTING ON PAGE 67

GUERILLA



For the C-64

By Cleveland M. Blakemore



CRYPTO-SECRET-CRYPTO
(BURN THIS MESSAGE AFTER IT HAS BEEN READ)

TO: RED LEADER TEAM
AT: FORTRESS
REGARDING: GREEN FORCES PENDING AGGRESSIVE ACTION.
FROM: RED TEAM COMMANDER

WARNING! WE HAVE RECEIVED ADVANCE SCOUT REPORTS OF VERIFIABLE ENEMY ACTIVITY IN YOUR AREA. WE CURRENTLY ESTIMATE THAT ANYWHERE BETWEEN 5 AND 15 BATTALIONS OF ENEMY TROOPS ARE EMPLACING OPERATIONS IN THE RUINS OF CIVILIAN BUILDINGS OPPOSITE THE WALL OF THE RED TEAM FORTRESS. WE ARE ONLY ABLE TO PROVIDE YOU WITH FOUR BOXES OF AMMUNITION AT THIS TIME, AS WE HAVE TO CONSERVE OUR OWN SUPPLY. YOU ARE TO HOLD THE FORTRESS AT ALL COSTS. THIS IS A DIRECT ORDER. THERE CAN BE NO WITHDRAWAL UNDER HOSTILE FIRE. WE EXPECT THAT THE GREEN TEAM WILL BEGIN THEIR ASSAULT ON YOUR CURRENT LOCATION AT DAYBREAK TOMORROW MORNING. EXPECT ENEMY INCOMING BY 0800. IT WILL PROBABLY GET PRETTY HAIRY DOWN THERE.
GOOD SHOOTING...AND GOOD LUCK!

---END OF MESSAGE---

We've never published a really gung ho military game in *Ahoy!*, so I figured it was time to do one. In this short and sweet BASIC-ML hybrid program, you fight off a communist horde who are launching a massive assault on your location. As they appear in the windows of the bombed out ruins on the other side of the stone wall of your fortress, you attempt to shoot them before they can get a bead on the wall with their LAWs (Light Antitank Weapons). If you are too slow, they fire a rocket which blows away a portion of the wall. The resulting damage is registered on the red bar at the bottom of the screen. When this bar vanishes, the fortress has been overrun.

The game uses a joystick in Port 2. The white cross represents your aiming point. Center this on a soldier as he appears in the window and press the button. If you hold the trigger down, the gun will fire continuously. This is very unwise, however, because you only have four boxes of ammo to begin with, and when they are used up, you're at the mercy of the GREEN aggressors.

The game has four different attack waves, with complete havoc breaking loose on the fourth one. Soldiers will be popping up in every window and firing everything they've got left to try to destroy the fortress. If you manage to survive this last attack, the RED army will be victorious over the GREEN forces. Otherwise, you'll end up reading propaganda and growing potatoes.

The soldiers are worth 150 points apiece, multiplied by the number of the wave. Any score over 70,000 is good.

It's very difficult to repel the GREEN forces. They're mean. They're hateful. They cross the street against the light. They've got tons of overdue library books. Just last week, they killed Rambo by inserting bullets into his body manually.

Good luck! ☐ *SEE PROGRAM LISTING ON PAGE 70*



IT TAKES A LOT TO IMPRESS ME.

BRIAN DOUGHERTY
Software Designer/CEO
Berkeley Softworks

And Q-Link, the dynamic telecommunications service for Commodore® owners, does just that!

As the developer of GEOS, the graphics environment operating system for Commodore 64s and 128s, I know how good Q-Link is. Personally and professionally. Here at my company, we recommend Q-Link to all our customers. We use it ourselves, too, for online customer service . . . so you can get help when you need it. We've also found it's a very efficient way to provide upgrades and patches and to announce new GEOS-compatible products.

With just your Commodore, a modem and Q-Link software, a new world of personal computing options opens up: thousands of programs you can download and keep; advanced graphics (thanks to GEOS); an exclusive help line to experts at Commodore; online educational courses taught by real teachers; fun; games; friendship, you-name-it. Q-Link lets you realize the full potential of Commodore computing—right at your fingertips.

Do I use Q-Link? Absolutely! And what's really impressive is that when you join Q-Link, you get a modem and Q-Link Software free! To my mind that's a tough offer to turn down. But don't take my word for it. Find out for yourself. Call . . .

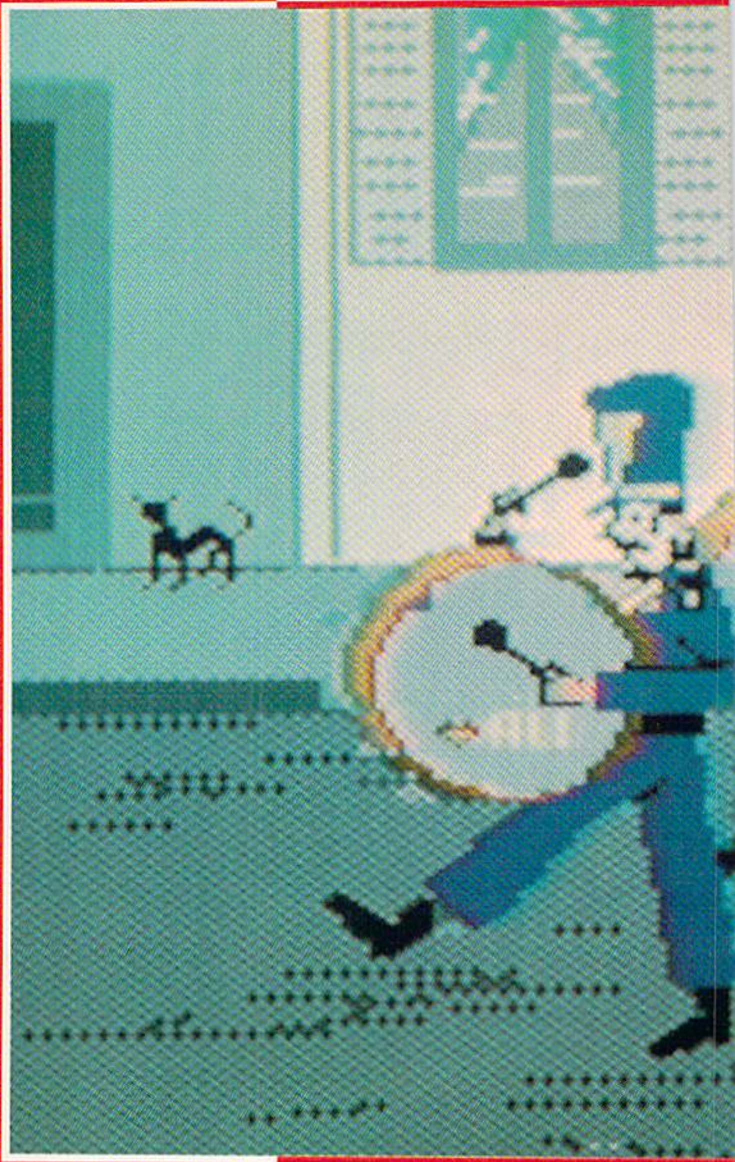
1-800-782-2278 ext. 1530

Call today to get your free modem and software.

Q-link℠

8619 Westwood Center Drive
Vienna, VA 22180

ART GALLERY

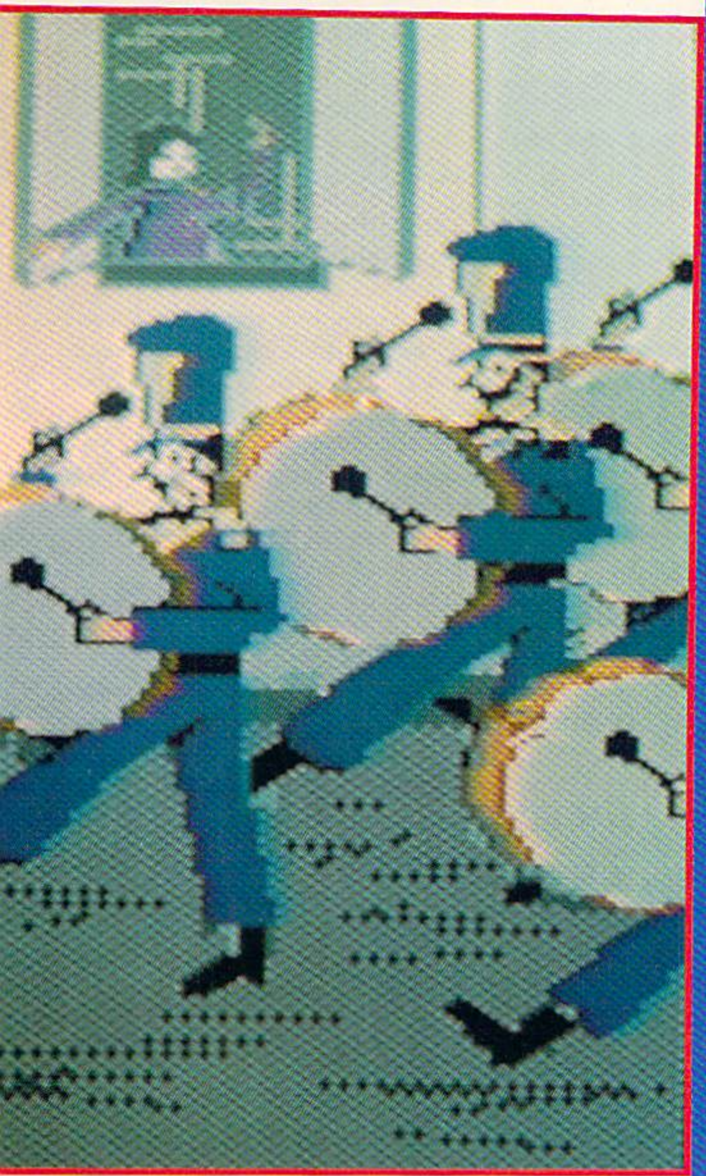


The *Ahoy! Art Gallery* offers the opportunity for fame and fortune to aspiring Commodore artists. Send your work on disk to *Ahoy! Art Gallery,* Ion International Inc., 45 West 34th Street–Suite 500, New York, NY 10001. Indicate the drawing package or file format of the images. Graphics produced on the Amiga are eligible for inclusion in *Ahoy!'s AmigaUser;* C-64, C-128, and Plus/4 images are eligible for inclusion in *Ahoy!* If your image is published, you will receive a free one-year subscription. Current subscribers will have their subscription extended by one year.

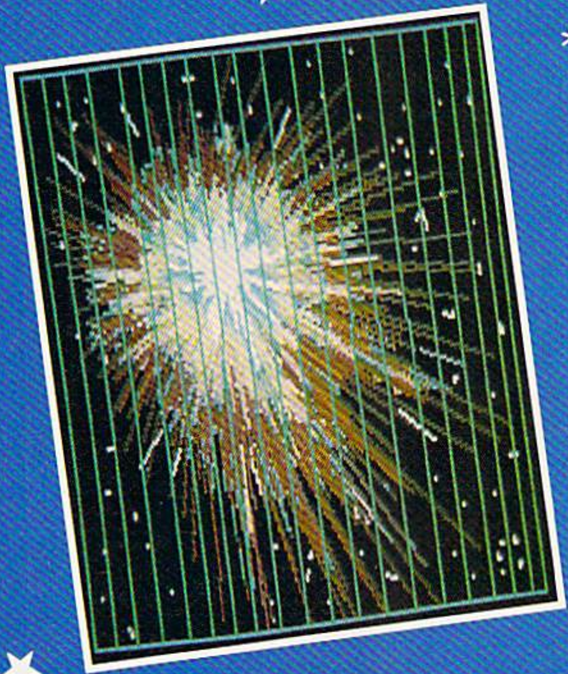
Note that the *Art Gallery* is not a contest. Published pictures are selected in an arbitrary and capricious fashion by the *Ahoy!* Art Director, based solely on their artistic merit.













Irony of ironies—only two Americans are among the contributors to this month's 4th of July edition of the *Art Gallery*. (Although, as the old joke goes, they have a 4th of July in other countries—they just don't celebrate Independence Day.) At left are two treatments of *Presidents' Day*—traditional and punk—by Michael Mikottis (Berwyn, IL). Above is *Marching Band* by Heinz Diekert (Vernon, BC), pounding out "Stars and Stripes Forever" just as surely as mom made little green apple pie (or something like that). At right is an unforgettable *Reagan* by Alberto Valsecchi (Milano, Italy), *Fireworks* by Tom Kane (Buffalo, NY), and *Post Office* by Robert M. Ellis (LaSalle, Quebec). True, it's a Canadian post office. But where would our neighbors north of the border be without the American Revolution? Still trapping furs, no doubt, and trading them to us for three-cornered hats and powdered wigs. Here's wishing all our readers a safe and enjoyable holiday.

PLASMATRON
Avantage
Commodore 64
Disk; $14.95

Plasmatron is a perfect example of what can only be called "generic entertainment software." These generally well-executed but hackneyed game programs are produced in Europe and sold in the U.S. by mainstream publishers. Often the American publisher uses a "discount" label name. In this instance, the Avantage banner signifies Accolade's lower-priced software.

European programmers grind out these rehashes of arcade "standards" like sausage. Games which should be the result of a creative process are instead formulaic and predictable.

On the other hand, as with all generic product, the user knows what he's getting and he gets it at a good price. The entertainment is based on play mechanics that are already proven successes, and the lower price gives gamers a chance to acquire playware at bargain rates.

Plasmatron is just such a piece. Originally produced by The Zen Room and brought to these shores by Avantage, it's a horizontally scrolling science fiction shootout set against a variety of scrolling backgrounds. The user pilots a Plasmatron fighter, a futuristic combat aircraft, over the planet Loughton 2, encountering fierce resistance from hostile forces. (Unfortunately, the skimpy instructions never specify whether these "aliens" are native to this world.) The objective is to blow up as many ships and other targets as possible, then make it back to base before the limited shields are exhausted.

The enemy ships in the initial attack wave are sitting ducks, but before long the opposition stiffens. Then the player is up against giant insects, floating asteroids, and multicomponent fighter ships which take up to five hits to eliminate completely.

The only remotely original aspect of *Plasmatron* is its visual presentation. At first look, it *seems* to be the familiar side perspective seen in earlier games like *Defender* (Williams) and *Super Cobra* (Stern). The designers, however, actually "tilted" the lower part of the landscape and added the Plasmatron fighter's shadow to help manufacture a 3-D look. This is a purely cosmetic innovation that has no impact on the game whatsoever. But when it comes to generic software, one takes one's innovation where one finds it.

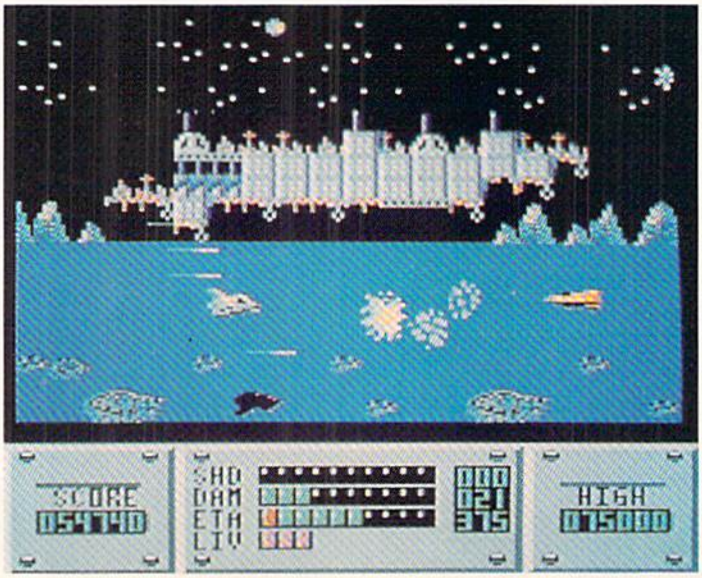
Plasmatron offers simulated three-dimensional graphics, but the inspiration for this pseudo-3D seems to be the old View-Master slide viewer system. When you looked into a View-Master, the world indeed had dimension, but only as a series of spaced backdrops. These backdrops were flat, like two-dimensional stage scenery. That's how *Plasmatron* looks. There's a horizon line about a third up the display area with backdrops slotted in sequence to create the illusion of depth.

The playfield also includes a console which displays the status of the ship's shields, damage, ETA, and lives remaining. It's difficult to speculate about the purpose of a damage reading on a ship which is destroyed as soon as its shields run out, especially since the instructions never even mention that there *is* a console! Damage is accumulated, however, even while shields remain intact, but the game plays at such a high speed it isn't possible to determine how it is being accrued. Current score and high score displays flank the console.

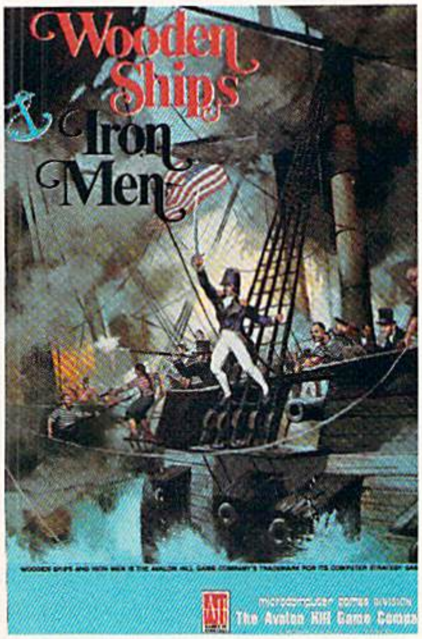
Plasmatron has some nice graphics and an original look, but there's nothing else here that hasn't been seen hundreds of times already.

Avantage/Accolade Software, 20863 Stevens Creek Blvd., #E, Cupertino, CA 95014 (phone: 408-446-5757).

—Bill Kunkel



Plasmatron's only somewhat original aspect is its visual presentation. The lower part of the landscape is "tilted" to help manufacture a 3-D look.
READER SERVICE NO. 151



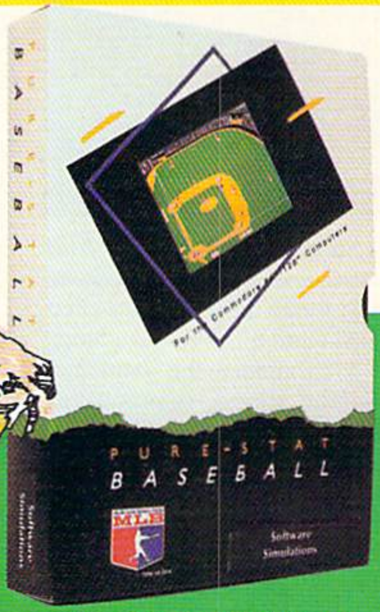
For the hard corps computer wargamer.
READER SERVICE NO. 152

WOODEN SHIPS & IRON MEN
The Avalon Hill Game Co.
Commodore 64
Disk; $35.00

Transferring a military simulation from the tabletop to the computer screen is always a chancy undertaking. Avalon Hill, which has a board game catalogue full of non-electronic classics, has tried numerous times with varying results.

PURE-STAT BASEBALL™





From **Software Simulations** comes an incredibly sophisticated, realistic, and complete statistical baseball simulation for zero, one or two players. Game play, manager's functions, graphics, and a complete statistical library establish Pure-Stat Baseball™ as the ultimate sports simulation program.

Suggested retail price

$39^95

Optional Team and Create Your Own Team Disks Available.

Available for C-64/128, Apple, IBM-PC

Game Play

Pure-Stat Baseball's underlying statistical framework simulates the realities of baseball like never before. Within this framework the program considers each player's batting statistics against both **LEFT**- and **RIGHT**-handed pitchers, where available, and pitchers stats vs. both **LEFT**- and **RIGHT**-handed batters. Every player's fielding and base-running abilities are also considered (an important factor when attempting to steal a base, etc.)

Select the team you'd like to manage, then pick the team you want to play against. Every team from the 1985 season is accurately represented, along with eight classic teams from the past. Determine your starting lineup, designate a starting pitcher, and make player substitutions when necessary. You call the plays, offense and defense. In a one-player game, your computer opponent displays un-canny intelligence in reacting to your managing decisions.

Manager's Functions

Built-in manager's functions provide extra versatility. The Trading function gives you the opportunity to create the "what if" team(s) you've always dreamed about. With this function you can have Dwight Gooden playing with Mickey Mantel and Babe Ruth. Or, if you like, you can form your own draft leagues from existing teams and play against your friends for your own pennant and World Series.

The Stat-Keeper function compiles all of the players' statistics for you and calculates Batting Averages and pitchers' ERAs. You can track your own teams' performances and print the season statistics for your club or league.

Stadium Disk

An optional Stadium Disk is also available that lets you play in any of the twenty-six Major League stadiums.

Statistics

Pure-Stat Baseball™ can maintain a complete statistical record of each team player's performance. You can print all of your players' year-to-date performance statistics at anytime throughout the season. All player stats and game Box Scores can be displayed on the screen or sent to an external printer for a hardcopy printout. The PRINT TEAM STATS function can be used at the end of the season to determine your Most Valuable Player, or to review your club's performance.

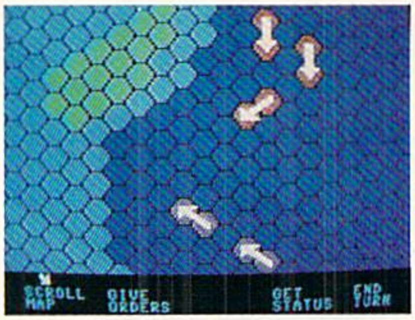
Pure-Stat Baseball™ also includes a unique Auto-Play option that lets the computer play a complete game in less than three minutes. A whole series of games can be played unattended, and an entire season of player and team statistics can be compiled over several days with all game stats printed out for your records.

Discover for yourself just how enjoyable a baseball simulation game can be. Pure-Stat Baseball,™ the first real baseball game on a computer.

See your local dealer, or call:

Software Simulations

959 Main Street, Stratford, CT 06497
203/377-4339



Wooden Ships ***has few computer frills.***

When S. Craig Taylor, Jr. unveiled the original edition of "Wooden Ships & Iron Men" under the Battleline Games imprint, strategy gamers hailed it as the greatest simulation of naval combat during the age of sail. Time has scarcely dimmed its reputation, so the release of the home computer disk ranks as one of the year's most important electronic gaming events.

Programmer Jim Jacob has executed a fairly literal translation of the board game. As a result, the Commodore edition of *Wooden Ships & Iron Men* makes virtually no use of the unique advantages of the computer. The sound is minimal, the graphics vaguely ape the counters and map of the board game, and gameplay is quite similar to Taylor's original creation.

If Jacob has transmitted the limitations of the board game to the computer version undiluted, the same can also be said of the simulation's strengths. The computerized *Wooden Ships & Iron Men* features the same wealth of detail, historical accuracy, and lively gameplay as its cardboard inspiration.

The main display of this one- or two-player contest is divided into two sections. The upper portion is a scrolling map with a movement grid. Each ship occupies two adjacent hexagons. Although the drawings are not especially detailed, players should have no trouble distinguishing friend from foe.

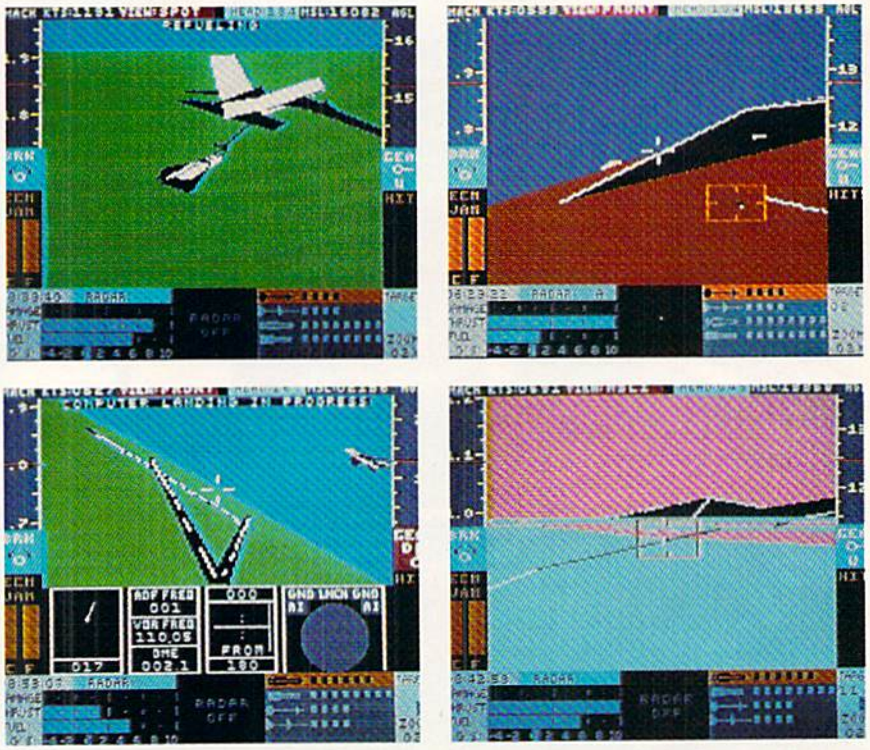
The lower portion of the screen presents option menus, ship status reports, and results of combat. The computerist employs the joystick to choose orders from menus and confirms them with a press of the action button. The program automatically puts a joystick-controlled cursor on the map when the player needs it for specific movement and firing commands.

Like most Avalon Hill creations, *Wooden Ships & Iron Men* subdivides turns into phases for ease of play. The sequence for a complete turn consists of the following phases: wind changes, unfoul tangled ships, movement, drop or raise anchors, grapple with nearby ships, boarding preparation, combat, melee combat and crew transfer, reload guns, and change sails.

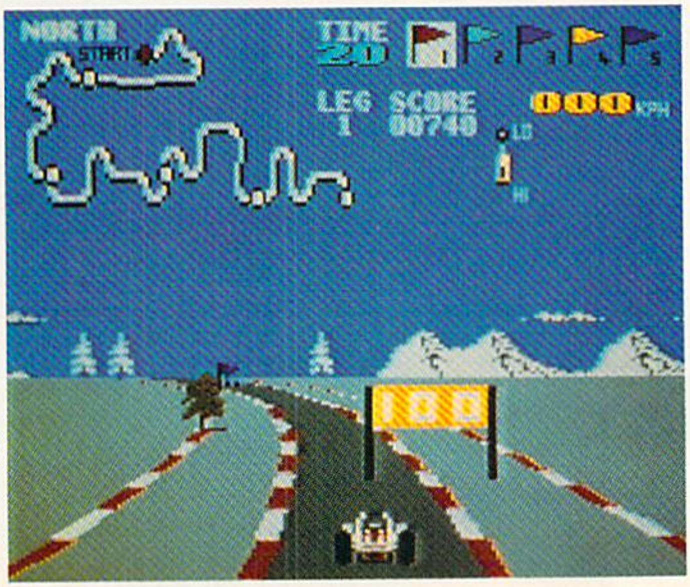
Although this list of procedures hints, correctly, that this is one complicated game, it's really not so daunting taken step by step. A mammoth 56-page manual contains exhaustive tutorials on both the routine of play and the construction of customized scenarios.

The ability to create additional battles is a great feature, especially for the naval historians among us, but the prepared scenarios are of paramount importance to most gamers. *Wooden Ships & Iron Men* really shines with 25 engagements drawn from the period from the American Revolution to the War of 1812.

The variety is incredible. The battles range from ship versus ship slug-



While* Stealth Mission *boasts remarkably accurate cockpit displays, its emphasis is on flight and combat elements. ***READER SERVICE NO. 153***



Speed Buggy ***offers five tortuous courses, all utilizing big rocks, wooden fences, and brick walls in various combinations as obstacles.*** ***READER SERVICE NO. 154***

fests like Constitution against Insurgent to huge multiship actions like the Battle of the Nile and Trafalgar. Most of the battles involve vessels from the US, France, and Britain, but there is some use of lesser powers like Spain and Venice when history so dictates.

For all its outstanding qualities, *Wooden Ships & Iron Men* appeals most strongly to hard corps computer wargamers. It minutely duplicates the board game, but does not add many of the trimmings non-wargamers have grown to expect from their software. Those who want an uncompromisingly authentic recreation of this colorful period in naval combat could well find a new favorite program once they set sail with *Wooden Ships & Iron Men*.

The Avalon Hill Game Co., 4517 Harford Rd., Baltimore, MD 21214 (phone: 301-254-9200). *–Arnie Katz*

SPEED BUGGY
Data East
Commodore 64
Disk; $29.95

To paraphrase an old saying, "God must love driving games. Otherwise he would not have made so many of them." It sometimes seems as though there's a new one every month, each only microscopically different from those which preceded it.

Speed Buggy, the latest title in this genre to reach market, breaks out of this automotive rut. It provides a totally original gaming experience that could never be confused with the sports car and race car simulations.

The gamer employs the joystick to control a four-wheel off-road vehicle. It turns on a dime, accelerates rapidly and, best of all, can skim along on two wheels if the player drives over one of the rocks which the designer has planted so helpfully in the middle of the road. After hours behind the wheel of electronic Indy cars and Corvettes, it's quite a shock to see the buggy launch into the air and sail over road obstacles.

The control scheme is simplicity itself. Moving the stick to the left or right governs lateral motion, slamming it forward puts the pedal to the metal, and pulling it back applies the brakes. The action button shifts between high and low gears. A rudimentary control panel in the upper right quarter of the screen presents all pertinent data, including the speed, lap number, current gear, and elapsed time.

The disk provides five tortuously twisted courses. The easiest is called "Off road," possibly because it has more hazards than the other four, which are labeled "North," "South," "East" and "West." Each has extensive scenery themed to its name. For example, "South" is a symphony of palm trees, sandy beaches, and pastel colors. All five utilize big rocks, wooden fences, and brick walls as the major obstacles, but the combinations vary greatly from course to course.

Speed Buggy is a race against time. The vehicle must pass the finish line before the countdown clock reaches zero. If the buggy navigates the entire course fast enough, the player earns the right to try another lap. The hazards are positioned differently for each lap, so there is pleasing variety even for the most skillful drivers.

The scoring system might dampen the spirits of a few novice drivers. Completing that first lap causes a tremendous increase in total score for most players. It is common to leap from a total in the 7000-8000 range to 30,000 in a single race just by beating the first time cutoff. So, until the computerist leaps that hurdle, be prepared for point totals to stay fairly static from round to round.

Like many other Data East productions, *Speed Buggy* bears the telltale traces of insufficient attention to detail. The program itself is fun and highly playable, but all the little things which enhance the gaming experience are either wrong or missing.

The implementation of the vanity board is a perfect example. Because *Speed Buggy* was obviously programmed for use with a tape drive, the disk does not retain high scores once the computer shuts down. There is no reason why this could not have been corrected before American republication. The same goes for the ludicrous title screen. It calls the game, presumably in an English transliteration of the original Japanese, "Buggy Goy." Insert snickers and ethnic jokes here.

Fortunately, none of these trivial flaws directly impacts the gameplay of *Speed Buggy*. It is tons of fun and a terrific addition to the Commodore 64's library of driving games.

Data East, 470 Needles Dr., San Jose, CA 95112 (phone: 408-286-7074). *–Arnie Katz*



Included among Stealth Mission's *eight scenarios are a mountain conflict, a battle at sea, and a bomber marathon.*

STEALTH MISSION
SubLOGIC
Commodore 64
Disk; $49.95

Stealth Mission is a new flight simulator from SubLOGIC. Roll it around in your mouth and say it again: "a new flight simulator from SubLOGIC." Sounds great, doesn't it?

SubLOGIC, in the person of Bruce Artwick, virtually invented flight simulation for microcomputers earlier in the decade. The original *Flight Simulator* (for the IBM PC) and the subsequent *Flight Simulator II* (other home systems) set the standards that dozens of other publishers spent years struggling to surpass. Secondary software, meanwhile, in the form of *FS* Scenery Disks, lets users take off and land their small aircraft anywhere from Chicago to Tokyo, and all points in between.

A few years later, SubLOGIC again blew the whole scene apart with *Jet*. This advanced simulator allows users to climb into the cockpit of several modern combat fighter jets and engage in mach-speed dogfights, aerobatics (stunt flying), and land on the deck of

Continued on page 51

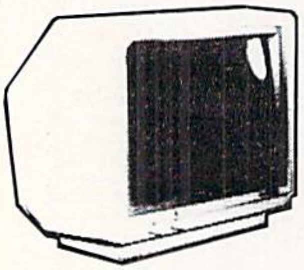
COMPUTER DIRECT

Will Not Be UNDERSOLD!

AND WE MEAN IT!*

Prices Expire 7-31-88

12" Monochrome Monitor



- High Resolution
- 80 Column
- 1000 Lines At Center
- Great For Business
- Fantastic Value

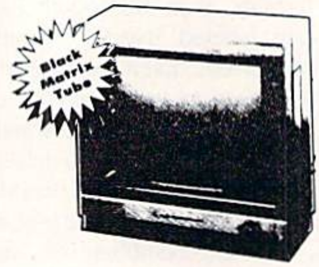
Our Low Sale Price

$79.95

List $329

(Add $10.00 shipping.*)

13" Color Composite Monitor



- Anti-Glare Screen
- Audio Speaker
- Volume Control
- Black Matrix Tube
- Horizontal Resolution 260 Lines
- Includes Composite Cable

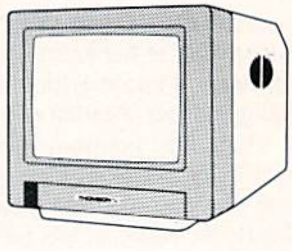
Our Low Sale Price

$169.95

List $329

(Add $14.50 shipping.*)

13" RGB/Composite Color Monitor



- Black Dot Screen
- 3 Monitors In One - Composite/RGB/80 Column
- Green Screen Only Switch
- Resolution 560 x 240 Pixels

Our Low Sale Price

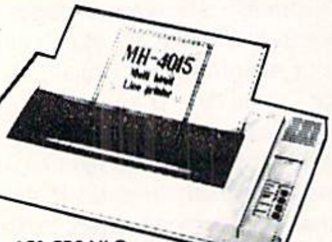
$219.95

List $399

(Add $14.50 shipping.*)

720 CPS 15" Printer

Multi-Head Wide Carriage Printer With High Dot Resolution



- 720 CPS - 150 CPS NLQ
- 8K Print Buffer
- Print Speed of 228 LPM In Draft
- 6 Print Heads Achieve High Dot Resolution In Higher Speed
- NLQ Speed Of 47.4 LPM
- 136 Column Wide Carriage
- IBM and Epson Mode
- Parallel Centronics Port

Our Low Sale Price

$799.95

List $1995

(Add $14.50 shipping.*)

5¼" Floppy Disks

Double Sided Double Density

Minimum Quantity 50
Maximum Quantity 250

19¢ each

(Add $4.00 shipping.*)

Box of 50 .... $9.50 (19¢ each)
(without sleeves)

Box of 50........... $10.50 (21¢ each)
(with sleeves)

Box of 100 (no limit) **... $24.00 (24¢ each)**
(with sleeves & labels)

Economy Sleeves	$1.00 per quantity of 50
Disk Labels	$3.95 per quantity of 100
Disk Notchers	$3.95 each

5¼" Sony Disks



49¢ each

Limited Quanties

(Add $4.00 shipping.*)

Single-Sided, Double Density
Sold In Packages Of 5 With Sleeves, Labels & Write Protects Included

$2.45 Per Box

THE VOICE MESSENGER

Speech Synthesizer for the Commodore 64

Features

- Infinite vocabulary
- Built-in software — nothing extra to buy
- Two voices each with intonation; adds character
- Simple BASIC commands for easy programmability
- Doesn't steal RAM from the BASIC workspace
- Won't stop screen action
- Clear, easy to understand speech
- Easy installation — just plugs in
- Keyvoicing — keys speak as they are pressed
- Direct text to speech capability

Our Low Sale Price

$29.95

List $49

(Add $3.00 shipping.*)

15 Day Home Trial • 90 Day Immediate Replacement Policy

COMPUTER DIRECT
22292 N. Pepper Rd, Barrington, IL. 60010
Call (312) 382-5050 To Order!

* Illinois residents add 6½% sales tax. All orders must be in U.S. Dollars. We ship to all points in the U.S., CANADA, PUERTO RICO, & APO-FPO. Please call for charges outside continental U.S. or C.O.D. MAIL ORDERS enclose cashier check, money order, or personal check. Allow 14 days delivery, 2 to 7 days for phone orders and 1 day express mail. Prices, specifications & availability subject to change without notice. Shipping and handling charges are not refundable. (Monitors only shipped in continental U.S.) COMPUTER DIRECT will match any valid nationally advertised delivered price on the exact same product with a comparable payment method (excluding any applicable sales taxes). A physical copy order of the current valid lower priced ad must be supplied with the order, or within 15 days of date of purchase.

VISA — MASTERCARD — C.O.D.

Reader Service No. 166

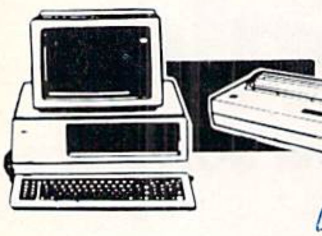
COMPUTER DIRECT

Will Not Be UNDERSOLD!

AND WE MEAN IT!*

Prices Expire 7-31-88

*15 MHz IBM® Turbo XT Compatible System



The Complete System Includes:

- 512K Professional Turbo XT Computer
- MS DOS 3.2 & GW Basic
- 12" Hi-Res 35 MHz Monochrome Monitor
- Genuine IBM® Printer With Interface & 1 Roll Of Heat Transfer Paper
- Word Processor, Spreadsheet & Data Base
- Complete DOS & System Tutor Included FREE!

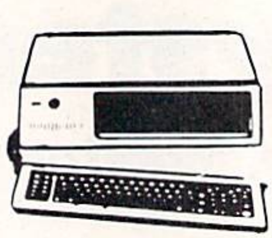
Our Low Sale Price

$699^95

List $1995

(Add $35.00 shipping.*)

*15 MHz IBM® XT Compatible Computer



- *15 MHz (Norton CI Rating)
- MS DOS 3.2 & GW Basic
- Parallel, Serial & Game Ports
- TTL Monochrome Monitor Card
- 512K Professional XT Turbo Computer
- Super Fast V20 CPU • "AT Style" Keyboard
- Security Keylock & Clock Calendar

Our Low Sale Price

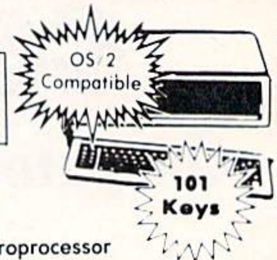
$599^95

List $1195

(Add $10.00 shipping.*)

12MHz 286 IBM® AT Compatible Computer

Requires MS DOS & GW Basic. $79.95 With Computer. Not Sold Separately.



- 80286 16 Bit Microprocessor
- 1.2 MEG Floppy Disk Drive
- Parallel & Serial Printer Ports
- 101 Key "AT Style" Keyboard
- BIOS w/Built-In Support For 3½" Drives
- Clock/Calendar & Security Keylock
- 0 or 1 Wait State

$899^95 With Floppy/Hard Drive Controller Card & 512K RAM Memory Installed

Our Low Sale Price

$599^95

List $2295

(Add $14.50 shipping.*)

Complete Commodore C128D System



The Complete System Includes:

- C-128D Computer With Built-In Disk Drive
- 12" Monochrome Monitor
- Genuine IBM® Printer With Interface
- One Roll Of Heat Transfer Paper
- C-128D Programmer's Reference Guide

Our Low Sale Price

$549^95

List $825

(Add $30.00 shipping.*)

Complete Apple® Compatible System



The Complete System Includes:

- Laser 128 Apple Compatible Computer
- Genuine IBM® Printer With Laser 128 Interface & 1 Roll Of Heat Transfer Paper
- 12" Hi-Resolution Monochrome Monitor
- Quality Word Processor, Spreadsheet And Data Base

Our Low Sale Price

$479^95

List $1409

(Add $27.50 shipping.*)

Ultra Hi-Speed Printer

300 CPS Draft • 50 CPS NLQ
With Color Printing Capabilities



- 300 CPS Draft - 50 CPS NLQ
- Front Panel Margin Setting
- Download Character Setting
- Variety of Characters & Graphics
- Parallel & Serial Interface Ports
- Auto Paper Load & Ejection
- IBM and Epson Mode
- Botton Feed & 10 K Buffer
- **Optional 7-Color Printing Kit ... Sale $99.95**

Our Low Sale Price

$279^95

List $499

(Add $10.00 shipping.*)

15 Day Home Trial • 90 Day Immediate Replacement Policy

COMPUTER DIRECT
22292 N. Pepper Rd, Barrington, IL. 60010
Call (312) 382-5050 To Order!

* Illinois residents add 6½% sales tax. All orders must be in U.S. Dollars. We ship to all points in the U.S., CANADA, PUERTO RICO, & APO-FPO. Please call for charges outside continental U.S. or C.O.D. **MAIL ORDERS** enclose cashier check, money order, or personal check. Allow 14 days delivery, 2 to 7 days for phone orders and 1 day express mail. Prices, specifications & availability subject to change without notice. Shipping and handling charges are not refundable. (Monitors only shipped in continental U.S.) **COMPUTER DIRECT** will match any valid nationally advertised delivered price on the exact same product with a comparable payment method (excluding any applicable sales taxes). A physical copy order of the current valid lower priced ad must be supplied with the order, or within 15 days of date of purchase.

VISA — MASTERCARD — C.O.D.

Reader Service No. 166

COMPUTER DIRECT
Is 1st In PRICE, SUPPORT, & GUARANTEE!
WE LOVE OUR CUSTOMERS!

Prices Expire 7-31-88

C64c Computer

Includes GEOS (Word Processor & Versatile Drawing Program)



Sale

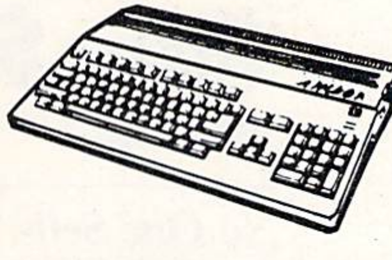
Our Low Sale Price

$149.95

List $249

(Add $10.00 shipping.*)

AMIGA



Amiga 500 Computer Sale

512K Computer Compatible with Amiga 1000 Software
List $799

Call For Price

Amiga 1010 External Drive Sale

3½" Compact-Size Micro-Disk Drive
List $299

Call For Price

Amiga 1084 Color Monitor Sale

13" RGB 80 Column x 25 Row, 640 x 400 Pixel
List $399

Call For Price

C128D Computer

Includes Full 128 Detached Keyboard & 1571 Drive With 128K Of User Memory Expandable To 640K. 3 Operation Modes (C64, C128 & CP/M, Ver. 3.0) 40/80 Column Output, 16 colors

Free Programmer's Reference Guide



Our Low Sale Price

$499.95

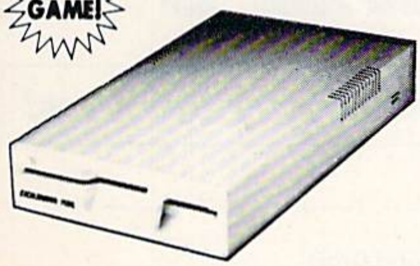
List $249

(Add $10.00 shipping.*)

Excelerator Disk Drive

Commodore Compatible
***Includes FREE Game**

FREE GAME!



Our Low Sale Price

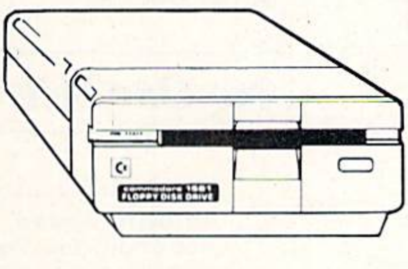
$149.95

List $249

(Add $10.00 shipping.*)

800K 3½" Disk Drive

Commodore 1581 Works With C64/C64c, and C128/C128D. Over 800K Bytes of Formatted Storage On Double-Sided 3.5" Microdiskettes.



Our Low Sale Price

$179.95

List $249

(Add $10.00 shipping.*)

300/1200 BPS External Modem

- 1200 Baud Bell 212A at 1200 or 300 Bits/Sec
- Fully Hayes Compatible
- Automatic Dialing & Answering
- Auto Redial At Receipt Of Busy Signal
- Built-In Speaker With Volume Control
- Built-In Help Commands
- Both Touch Tone and Pulse Dialing
- Uses RS-232 Interface (RS-232 Interface Cable Required

Sale Price $79.95

(Add $3.00 shipping.*)

List $139

300/1200/2400 BPS External Modem

- Fully Hayes Compatible
- Both Tone & Pulse Dialing
- Built-In Speaker With Volume Control
- Command Buffer That Holds Up To 40 Characters
- 2400 Baud Bell 212A At 300, 1200 & 2400 BPS
- Uses RS-232C Interface (RS-232C Interface Cable Required)

Sale Price $159.95

(Add $3.00 shipping.*)

List $249

15 Day Home Trial • 90 Day Immediate Replacement Policy

Call (312) 382-5050

We Love Our Customers!

Mail

COMPUTER DIRECT
22292 N. Pepper Road
Barrington, IL. 60010

Reader Service No. 166

COMPUTER DIRECT

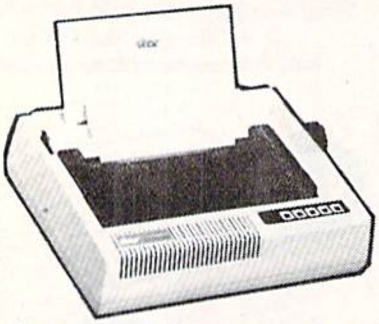
Will Not Be UNDERSOLD!

AND WE MEAN IT!*

Prices Expire 7-31-88

Super Hi-Speed 200 CPS Printer

Star Micronics LV-2010 10" Printer With Near Letter Quality



Sale $199.95

(Add $10.00 shipping.*)

List $499

No One Sells This Printer For Less!

- High Speed 200 CPS Draft
- Near Letter Quality
- Tractor and Friction Feed
- Automatic Sheet Feed Loading
- Both Parallel & Serial Interface Ports
- Continuous Underline
- IBM Compatible
- Ultra High-Res Bit Image Graphics

Printing Speed
200 Characters Per Second Draft
45 Characters Per Second NLQ

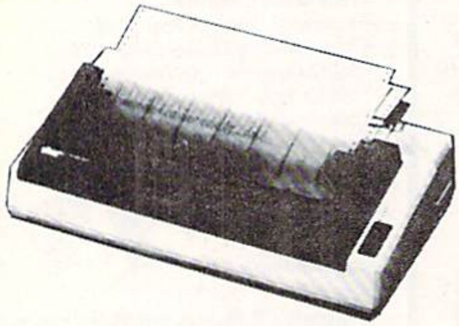
Print Method
Serial Impact Dot Matrix

Print Buffer
16K Bytes

Port
Both Centronics Parallel & Serial Ports

15" High Speed 160 CPS Printer

Splendid Mixture Of High Speed And Near Letter Quality



Sale $229.95

(Add $14.50 shipping.*)

List $499

No One Sells This Printer For Less!

- NLQ and High Draft Speed
- Continuous Underline
- Skip Over Perforation
- IBM Compatible
- Friction and Tractor Feeds
- Both Parallel & Serial Interface Ports
- High Res Graphics Printing

Printing Speed
160 Characters Per Second Draft

Print Method
Serial Impact Dot Matrix

Paper Feed
Sprocket or Friction Feed

Port
Both Centronics Parallel & RS-232 Serial Ports

15 Day Home Trial • 90 Day Immediate Replacement Policy

COMPUTER DIRECT
22292 N. Pepper Rd, Barrington, IL. 60010
Call (312) 382-5050 To Order!

* Illinois residents add 6½% sales tax. All orders must be in U.S. Dollars. We ship to all points in the U.S., CANADA, PUERTO RICO, & APO-FPO. Please call for charges outside continental U.S. or C.O.D. **MAIL ORDERS** enclose cashier check, money order, or personal check. Allow 14 days delivery, 2 to 7 days for phone orders and 1 day express mail. Prices, specifications & availability subject to change without notice. Shipping and handling charges are not refundable. (Monitors only shipped in continental U.S.) **COMPUTER DIRECT** will match any valid nationally advertised delivered price on the exact same product with a comparable payment method (excluding any applicable sales taxes). A physical copy order of the current valid lower priced ad must be supplied with the order, or within 15 days of date of purchase.

VISA — MASTERCARD — C.O.D.

Reader Service No. 166

COMPUTER DIRECT

Is 1st In PRICE, SUPPORT, & GUARANTEE!

WE LOVE OUR CUSTOMERS!

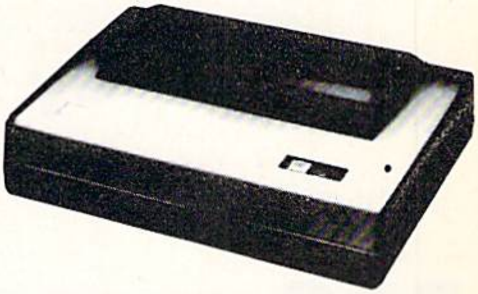
Prices Expire 7-31-88

Genuine IBM® 8½" Printer

8½" Letter Size • 80 Column • Limited Quantities

Sale $44⁹⁵

(Add $7.50 shipping.*) List $199



No One Sells This Printer For Less!

SAVE

- Unbelievable Low Price
- Advanced Dot Matrix - Heat Transfer
- Upper & Lower Case (With True Lower Descenders)
- Underline • Enlarged • Graphics
- Ready to Hook Up To Serial Port Of IBM® PCjr
- Low Cost Adapters For IBM®, Apple, Commodore & Laser Computers

Hi-Speed NLQ 180 Printer

Superb Near Letter Quality • Lifetime Warranty*

Sale $179⁹⁵

(Add $10.00 shipping.*) List $499



No One Sells This Printer For Less!

Price Break Thru

- High Speed Dot Matrix
- Near Letter Quality Selectable From Front Panel
- Italics - Elite - Condensed - Pica
- Super Graphics - 8K Buffer
- Tractor/Friction Feed
- Centronics Parallel Port
- * Lifetime Warranty On Print Head
- 6 Month Replacement Policy For Printer

15 Day Home Trial • 90 Day Immediate Replacement Policy

Call **(312) 382-5050** Mail

We Love Our Customers!

COMPUTER DIRECT
22292 N. Pepper Road
Barrington, IL. 60010

Reader Service No. 166

VARI-SCAN

For the C-64 and C-128

By Buck Childress

Well people, I'm about to hand down a secret recipe for the best RAM chip soufflé this side of Pluto. Just add a head full of frustration, a dash of space bar, and a couple of cursor keys to taste. Mix well, then jump up and down on Ol' Reliable till your anxiety is overtaken by grief. Heck, you've just hurled your trusty companion up to that Great Microchip in the Sky.

Of course I'm just kidding...I think. At any rate, we all get overwhelmed by frustration from time to time. After all, *some* of us are only human. Well, I suppose an explanation is in order.

Variables (or should that be frustration-ables?) have got to be one of the biggest contributors to premature hair loss, fried brain cells, and unintelligible babbling amongst computer programmers. Have you ever been in the process of writing a program, added a new (or so you thought) variable to the list, and then watched in total disbelief as your masterpiece took a nose dive? Shucks, if you'd only known that not-so-original variable was in there, you probably wouldn't be dangling from the ceiling right now. Why don't we get you off that ceiling (how are you going to explain the footprints to the landlord?) and lower the old blood pressure a bit. Maybe with a little bit of practice you'll be able to talk again. As for me, my brain cell just multiplied. With some luck both of them might do it again. Then I'll have four.

There have been programs written that will give a list of variables currently in the program. They work well, but you have to trace through the list to see if the variable you want to use is there. That's okay so long as the program isn't too long and you're not in any kind of hurry. But, when you get on a roll and your creativity and fingers are flying like a Concorde jet, those programs just won't do. So... *Vari-Scan* to the rescue. When you want to find out if a variable is in use, just type it, press RETURN, and bingo. You'll know in less time than it takes to yawn.

Vari-Scan automatically searches out any variable you choose, whether string, numeric, integer, or array. It's easy to use, and gets the job done pronto. It works equally well on both the C-64 and C-128.

After saving a copy of *Vari-Scan*, run it. The loader POKEs the machine language data into memory and checks for errors. When it's done you can activate *Vari-Scan* by typing SYS 52000 for the C-64, or SYS 4864 for the C-128, and then pressing RETURN.

Whenever you want to see if your program contains a certain variable, just type the variable name and press RETURN. If your chosen victim is in the program, *Vari-Scan* returns the line numbers that have it. They can't run and they can't hide. You've got 'em now.

If the variable you're searching for is an array, you have a couple of options. You an scan for an exact match by entering the entire name such as A(1). A(1) would have to be in the program for a match to occur. But, if you want to know if the variable A is used in any arrays (e.g., A(J), A(1,5), A(2,X,7)), enter the variable name like this:

```
A(£
```

The LIRA (£) sign tells *Vari-Scan* to list all occurrences of A as used in arrays.

Vari-Scan can discern the difference between all variables. For example, let's say that you want to search for the variable A1. *Vari-Scan* will only scan for the numeric variable A1. It will not report any integer, string, or array variables of the same name, nor will it inadvertently report a variable containing an A, such as AB. The same holds true for other types of variables. *Vari-Scan* won't scan for anything in quotation marks, or on a line following a REM or DATA statement.

Vari-Scan checks for extended variable names. If you've used HOME$ in a home budget program, *Vari-Scan* recognizes it as HO$ (the same as the 64 and 128 does). So you don't have to worry about some weird variable sliding by your prying eyes.

Vari-Scan can be deactivated by pressing the BACK ARROW (←) key. SYS 52000, or 4864, to reactivate it.

The next time you need an instant variable check, give *Vari-Scan* a try. It's easy to use, gets the job done fast, and keeps variable frustration at a minimum. Besides, who wants to do a tap dance on the keyboard? Then you couldn't use it as a frisbee. □

SEE PROGRAM LISTINGS ON PAGE 64

TEVEX Computer Software 1-800-456-1162

SSI

	List $	Our $
B-24	$35	$24
Battlecruiser	$60	$41
Eternal Dagger	$40	$28
Gettysburg	$60	$41
Kampfgruppe	$60	$41
Panzer Strike	$45	$31
Phantasie III	$40	$28
President Elect	$25	$18
Questron II	$40	$28
Realms Darkness	$40	$28
Roadwar 2000	$40	$28
Roadwar Europa	$40	$28
Shard of Spring	$40	$28
Shiloh	$40	$28
Sons of Liberty	$35	$24
Wargame Constr.	$30	$21
War South Pacific	$60	$41
Wizard's Crown	$40	$28

MICROPROSE

	List $	Our $
Airborne Ranger	$35	$24
F-15	$35	$24
Gunship	$35	$24
Pirates	$40	$28
Silent Service	$35	$24
Stealth Fighter	$40	$28

EPYX

	List $	Our $
California Games	$40	$28
Champ. Wrestling	$40	$28
Destroyer	$40	$28
Movie Monster	$20	$15
Spy vs Spy III	$25	$18
Street Sports	$40	$28
Sub Battle	$40	$28
The Games	$40	$28
World Games	$40	$28
EPYX JOYSTICK	$22	$15

SSG

	List $	Our $
Battlefront	$40	$28
Battles Civil War	$40	$28
Battles Normandy	$40	$28
Carriers at War	$50	$34
Europe Ablaze	$50	$34
Halls Montezuma	$40	$28
Reach - Stars	$45	$31
Russia	$40	$28
Run 5 Magazine #8		$4

ACCOLADE

	List $	Our $
Apollo 18	$30	$21
Card Sharks	$30	$21
4th & Inches	$30	$21
Hardball	$30	$21
Plasmatron	$15	$12
Power at Sea	$30	$21
Spy vs Spy I & II	$15	$12
Test Drive	$30	$21
The Train	$30	$21

ACTIVISION

	List $	Our $
Aliens	$35	$24
Champ Basketball	$35	$24
Champ Football	$35	$24
Labyrinth	$30	$21
Last Ninja	$35	$24
Maniac Mansion	$35	$24

INFOCOM

	List $	Our $
Beyond Zork - 128k	$45	$31
Border Zone	$35	$24
Lurking Horror	$35	$24
Nord & Bert	$35	$24
Sherlock	$35	$24
Stationfall	$35	$24

AMIGA

	List $	Our $
Allen Fires	$40	$28
Balance of Power	$50	$34
Breach	$40	$28
Breach Scenario	$25	$18
Defender - Crown	$50	$34
Ferrari Formula 1	$50	$34
Fire Power	$25	$18
Hunt-Red October	$50	$34
Jet Simulator	$50	$34
Kampfgruppe	$60	$41
King of Chicago	$50	$34
Return to Atlantis	$50	$34
Roadwar 2000	$40	$28
Roadwar Europa	$45	$31
S.D.I.	$50	$34
Silent Service	$40	$28
Sinbad	$50	$34
Starfleet I	$55	$38
Terrorpods	$40	$28
Test Drive	$45	$31
Weaver Baseball	$50	$34
Winter Games	$40	$28

ELEC. ARTS

	List $	Our $
Adv Constr. Set	$15	$12
Age of Adv	$15	$12
Amnesia	$40	$28
Arcticfox	$33	$23
Bard's Tale I or II	$40	$28
Chess 2000	$40	$28
Chuck Yeager	$35	$24
Dragon's Lair	$25	$18
EOS	$30	$21
Legacy - Ancients	$30	$21
Lords - Conquest	$15	$12
Marble Madness	$30	$21
Pegasus	$30	$21
Skate or Die	$30	$21
Seven Cities	$15	$12
Skyfox II	$30	$21
Strike Fleet	$30	$21

ORIGIN SYS.

	List $	Our $
Auto Duel	$50	$34
Moebius	$40	$28
Ogre	$30	$21
Ultima I	$40	$28
Ultima III	$50	$34
Ultima IV	$60	$41

ETC.

	List $	Our $
Alt Reality - City	$30	$21
AR - Dungeon	$40	$28
Blue Powder	$50	$34
Borodino	$60	$42
Choplifter-Magic	$15	$12
Dark Horn	$35	$24
Defender - Crown	$35	$24
Echelon	$45	$31
Full Count Bball	$40	$30
Gauntlet	$35	$24
Guderian	$30	$21
High Seas	$50	$34
Might & Magic	$40	$28
ML Baseball	$40	$28
ML Wrestling	$30	$21
NBA	$40	$28
Starfleet I	$40	$28
Stealth Mission	$50	$34
Up Periscope	$30	$21
Wizardry I	$40	$28
WC Leader Board	$40	$28
Wooden Ships	$35	$24
Wrath - Denethenor	$20	$15

EDUCATIONAL

	List $	Our $
Carmen - U.S.	$40	$28
Carmen - World	$35	$24
Early Games	$35	$24
Easy as ABC	$40	$28
Facemaker	$21	$15
Fraction Factory	$30	$21
Kindercomp	$21	$15
Math Blaster	$50	$34
Piece of Cake Math	$35	$24
Reader Rabbit	$40	$28
Rocky's Boots	$35	$24
Speed Reader II	$50	$34
Spell It	$50	$34
SB Spellgrabber	$30	$21
Stickybear ABC	$30	$21
SB Math II	$30	$21
SB Numbers	$30	$21
SB Opposites	$30	$21
SB Reading	$30	$21
Word Attack	$50	$34

PRODUCTIVITY

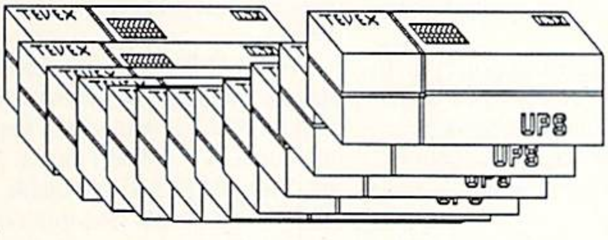
	List $	Our $
Certificate Maker	$40	$28
GEOS - 64	$60	$41
GEOS - 128	$70	$47
Geos Desk Pack I	$35	$24
GeoCalc	$50	$34
GeoDex	$40	$28
GeoFile	$50	$34
GeoProgrammer	$70	$47
GeoPublish	$70	$47
GeoSpell	$30	$21
GeoWrite Wkshop	$50	$34
Newsroom	$50	$34
Outrageous Pages	$50	$34
Paper Clip III	$50	$34
Print Shop	$45	$31
Print Shop Comp.	$35	$24
Toy Shop	$30	$21

ACCESSORIES

Sony DSDD 5.25	$9
Tevex DSDD w/library case	$8
Sony DSDD 3.5	$20
3-M DSDD 3.5	$22
Cleaning Kit 5.25	$6
Cleaning Kit 3.5	$6
EPYX 500 Joystick	$15
Mouse Pad	$6
Safe Strip 6 Surge Protector	$22
Storage Case 60 5.25	$9
Storage Case 40 3.5	$9
Universal Printer Stand	$22

Same Day Shipping

Just call before 3:30 and we'll ship your order today by UPS. Your package is only days away with Tevex.



2 for 1 SALE! Buy 2 get 1 FREE for selected Electronic Arts software. Call for details.

CALL TOLL-FREE 1-800-456-1162

FREE CATALOG WITH YOUR FIRST ORDER

Open 9-8 Mon. - Fri. 10-5 Sat.

Retail Store open same hours.

Georgia residents call 404-934-5059. When ordering by mail send money order. Include phone number. SHIPPING: Add $3.00 for shipping and handling charge. Georgia residents add 4% sales tax. Shipping for Canadian orders is 5% of order, with a $4.00 minimum. U. S. Mail, APO & FPO orders add 5% of order, with a $4.00 minimum. Shipping for all other foreign orders is 15% of order, with a $10.00 minimum. All sales are final.

New Titles are underlined

TEVEX

4205 First Ave, Suite 100
Tucker (Atlanta), GA 30084
404-934-5059

VISA MasterCard

Reader Service No. 156

Thermal Connection

Real-World Interface: Temperature Sensing

The computer can do much more than merely execute programs. When connected to the proper devices, the computer can measure various characteristics of the real world (the world outside of the computer's circuitry), and it can control some of those characteristics. In previous articles we have connected the computer to photo-cells, light-emitting diodes, potentiometers, and relays. This month we will use a temperature-sensing device, a thermistor, to turn the computer into a moderately accurate digital thermometer.

Both the C-64 and the C-128 have built-in analog-to-digital converters. These are called "A to D" or simply "A/D" converters. The function of an A/D converter is to receive an analog input signal and to convert it to a digital value. Specifically, the A/D converters in the Commodore computers give an integer value from 0 to 255, which is proportional to the amount of resistance applied to their inputs.

A TO D FUNDAMENTALS

If you apply a short circuit between the 5 volt supply voltage and the A/D's input, the A/D sees a resistance of zero ohms and converts this to a digital value of 0. If you leave the input to the A/D converter unconnected or open, the A/D sees essentially an infinite resistance between its input and the 5 volt supply. Then the A/D converter gives the largest value it can, namely 255.

The A/D converters in the Commodore computers give an output value of 1 for approximately every 10,000 ohms of input resistance. That is, 50,000 ohms corresponds to an A/D output of roughly 50. This is fairly accurate for resistance below 100,000 ohms and output values of less than 100.

Because of different A/D input circuitry, the C-64 reaches its maximum value of 255 with an input resistance of roughly 500,000 ohms, whereas the C-128 gives a maximum output of 255 with an input of about 250,000 ohms. Note that these are only "rules of thumb." Any serious application requiring actual resistance values must calibrate the A/D converters. If you need only relative values ("is the paddle turned more to the left or more to the right?", for example), calibration may not be necessary.

Just to eliminate any confusion, I should mention that A/D converters generally convert analog input voltages (not resistance) into digital quantities. (Refer to *Analog to Digital Adventures*, October 1986 *Ahoy!*, for further discussion of A to D conversion.) Since voltage and resistance are related, it is appropriate and more useful to talk about input resistance in this application.

CONVERSION SOFTWARE

The A/D converters are accessed through Control Ports 1 and 2 (the joystick ports) on the right side of the computer. Each port can handle two resistance inputs. Normally game paddles are plugged into the A/D pins of these ports. There are only two A/D converters in the Commodore computer, but there is an electronic switch which can select the inputs from either Port 1 or Port 2. That way two A/D converters take care of four analog inputs.

BASIC 7.0 in the C-128 uses the POT command to read the A/D converters. POT(1) and POT(2) give values corresponding to Control Port 1 inputs (closest to the front of the computer). POT(3) and POT(4) correspond to Control Port 2.

For the C-64, you must read the A/D converters by other means. On page 346 of the *C-64 Programmer's Reference Guide (PRG)* is a machine language program for reading all four A/D ("paddle") inputs. It states that reading the paddles from BASIC is not reliable. The machine language program *C-64 Paddle Routine* on page 66 of this magazine is a condensed version of the *PRG* program which allows inputs only in Control Port 1.

The procedure to perform an A/D conversion and to read the results is as follows:

1. Set the electronic switch at address $DC02 (addresses

By Dale Rupert



are in hexadecimal) to enable Port 1 or Port 2 (or both) inputs to the A/D converters.

2. Wait briefly for the inputs to be converted.

3. Read the outputs of the two A/D converters at $D419 and $D41A.

The remarks at the beginning of the machine language program provide more details. Note that interrupts must be disabled. The Complex Interface Adapter (CIA) chip has pins which can be inputs or outputs. Two of its pins (PA7 and PA6) are tied to the electronic switch. The Data Direction Register (DDR) of the CIA at address $DC00 must be programmed so that those two pins are outputs. Then 1's are written to those two pins to close the switches and bring the Control Port paddle signals to the A/D converters.

This machine language program enables the inputs from both Port 1 and Port 2, although only Port 1 will be used. The A/D converters are part of the Sound Interface Device (SID) chip. They continuously convert whatever input is available. If nothing is plugged into either control port, the A/D converters see infinite resistance and give values of 255.

Although the *PRG* says BASIC is not reliable, I have found that these statements work nearly as well as the machine language routine:

```
PRINT PEEK(54297)

PRINT PEEK(54298)
```

They read the SID chip's A/D registers at $D419 and $D41A. The first address is for the A/D converter for input resistance between pin 7 (5 volts) and pin 9 (POT X) of the control port connector. The second address is the A/D converter which measures resistance between pin 7 (5 volts) and pin 5 (POT Y). Here "POT" stands for "potentiometer," which is a variable resistor.

HARDWARE PREPARATION

You need less than $5 worth of parts to convert your computer into a digital thermometer:

1. Subminiature D 9-pin female connector (Radio Shack 276-1538 or equivalent).

2. Thermistor (Radio Shack 271-110 or any other with at least 10,000 ohms at 25 degrees C).

3. Wire: any length, 20 to 24 gauge.

4. Epoxy or waterproof glue.

Solder two equal lengths of wire (I used about four feet for each) to pin 9 and pin 7 of the 9-pin connector. Pin 9 is the POT X (paddle) input, and pin 7 is 5 volts. For convenience, you may twist the wires together.

Thermistors come in a variety of sizes and shapes. The one listed above is a small bead about the size of a pencil tip. Thermistors have two wire leads attached to them. The two leads of the thermistor are soldered to the other ends of the wire. It doesn't matter which wire goes to which lead of the thermistor. The schematic of the finished apparatus is in Figure 1.

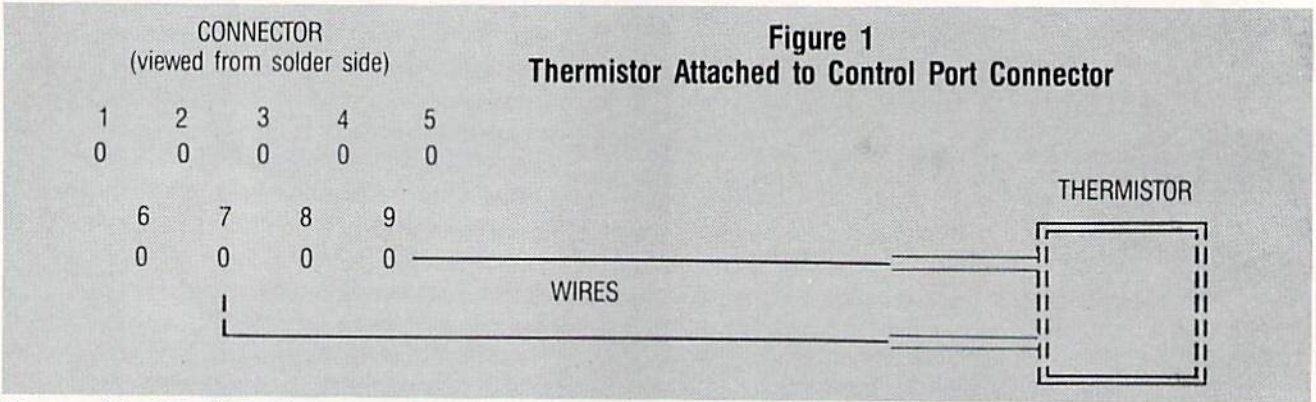
Before using the thermistor with the program we will discuss, you must make its leads waterproof. Daub some epoxy, silicone glue, or other waterproof glue around the leads of the thermistor and the ends of the wire to which they are soldered. Don't coat the body of the thermistor itself any more than necessary to seal the leads. (Be sure the leads are not touching each other.)

PLUG IT IN

After constructing the thermistor apparatus, plug it into Control Port 1 (the joystick port on the right side of the computer, the one closer to the front of the computer).

Remember that computer chips connected to pins on the Control Port connectors are static-sensitive. Chips in the computer can be damaged by electrostatic discharge. Discharge yourself by touching something metal before connecting or disconnecting anything (joysticks included) at these connectors. Don't use your computer in a very dry, static-prone environment without proper precautions.

Figure 1
Thermistor Attached to Control Port Connector



Turn on the computer and enter this quick program:

C-64 VERSION:

```
10 PRINT PEEK(54297), : GOTO 10
```

or

C-128 VERSION:

```
10 PRINT POT(1), : GOTO 10
```

You should see a series of very close numbers, typically about 15 for the thermistor listed above at room temperature (70 degrees Fahrenheit). Those numbers are the digital outputs from the A/D converter as it measures the resistance of the thermistor.

If you get very erratic numbers, or all zeros, or all 255's, then something is wrong. All zeros means that either the

wires or thermistor leads are shorted together (touching). Look closely at both ends of the wires and check the connector for solder bridges.

All 255's means that either the wires are not well-soldered to the connector or to the thermistor, or the connector is not plugged into the computer connector firmly, or the thermistor is open-circuited (faulty), or the connector is plugged into the wrong control port.

If you see very random numbers, make sure you are plugged into Port 1. Turn your computer off, then back on, and reenter the program (make sure some other programs in the computer are not interfering with the thermistor–this would be a problem with only the C-64 version).

THE THERMISTOR

If all went well, the number repeatedly displayed is a digital representation of the thermistor's resistance. Squeeze the thermistor in your hand. You should see the value on the screen change. It should decrease since the resistance of a thermistor decreases as the temperature rises. Notice that the value on the screen gradually changes to the new value. It takes a while for the thermistor to change temperature.

The advantages of the thermistor for our application are that it is a resistive device (unlike a thermocouple which is a voltage device), it is inexpensive (unlike a platinum Resistive Temperature Device or RTD), and it has a relatively large thermal sensitivity (its resistance changes by a factor of ten or more for a one hundred degree temperature change).

The disadvantage of a thermistor is that it is very nonlinear. This means that its resistance does not change uniformly with temperature. In fact, the relation between resistance and temperature for a thermistor is given by this equation:

$$1/T2 = 1/T0 + 1/B * LN(R2/R0)$$

If the thermistor's resistance R0 is known at a certain temperature T0, and if a calibration constant B is known for the thermistor, then any other temperature T2 may be calculated from the above formula by measuring R2, the thermistor's resistance at that temperature. Note that LN is the natural logarithm function given by LOG in BASIC. Also note that temperatures must be given in degrees Kelvin. More on that later.

As we have seen, our computer has the ability to measure the thermistor's resistance. We will also let the computer perform the calculations in the formula to tell us the new temperature.

CALIBRATION CAN BE FUN

We must measure the thermistor's resistance at two different *known* temperatures in order to calculate B, the calibration constant. The procedure we will use is this:

1. Measure resistance R1 at room temperature T1.
2. Measure resistance R0 at ice water temperature T0.
3. Calculate B, derived from the previous equation, since T1 and T0 are known:

$$B = LN(R1/R0) / (1/T1 - 1/T0)$$

You could use any two temperatures, but these are easy to obtain.

Once the computer determines the calibration constant

B, the thermistor can be used to measure any other unknown temperatures. The computer will determine any new temperature T2 with this procedure:

1. Measure resistance R2 at the new temperature.
2. Calculate the new temperature T2:

$$T2 = 1 / [(1/T0) + (1/B) * LN(R2/R0)]$$

Notice that since this formula depends upon the ratio of the two resistances, it doesn't matter what units they are measured in as long as they are the same. We will use arbitrary Commodore A/D converter units for measuring resistance. Temperatures must be in degrees Kelvin.

PUT IT ALL TOGETHER

If all the preceding has seemed too complicated, fear not. The program *Thermal Connection* on page 66 takes care of all the difficulties. Before you run this program you must

1) find out what your room temperature is (use a thermometer or look at your thermostat) and

2) put some cold water along with some ice cubes or crushed ice into a dish and set it where your thermistor plugged into the computer can reach it.

Plug the thermistor into Control Port 1 and run the program. Note that C-64 users must change lines 100, 1020, and 2030 as shown in lines 5001-5003. Also, C-64 users must add line 11 (shown in line 5000) and add lines 5010-49184. These changes for the C-64 replace the POT(1) function of the C-128.

The program asks if you already know the calibration constants for the thermistor. If you reply "N" to indicate that you don't know them, the program starts the calibration process beginning at line 3070. You must have the thermistor at room temperature (or any other known temperature except the freezing point of water). Enter that temperature in degrees Fahrenheit. (The computer will convert it to degrees Kelvin in line 3180.)

The program calls the routine at line 1010. You will see a display on the screen similar to the earlier test of the thermistor. Once the numbers appear to have settled to one or two values, press any key. The program calls the routine at line 2010 to read and average the next ten resistance readings. This value is R1, which corresponds to your room temperature T1.

Now you are instructed to put the thermistor into the ice water. The computer assumes its temperature T0 is 32 degrees F. Press any key to see the new A/D values on the screen. You should see them gradually increase until they reach some steady level. Again press any key once the numbers appear to have settled to one or two values. The computer averages the next ten readings and calls the result R0, corresponding to T0.

The computer has all the information to calculate the calibration constant B in line 3200. The values of B, T0, and R0 are then shown on the screen. You should write them down. That way, the next time you run the program you can enter them manually rather than going through the calibration process again. (My values with the listed thermistor are typically B=2884, T0=273, and R0=29. Room temperature T1 of 293 degrees K (68 degrees F) gives an A/D reading R1 of 15.)

Press any key once you have recorded the calibration constants. You are returned to the main loop at line 100 which repeatedly reads the thermistor's resistance R and converts it to a temperature T in degrees Kelvin in line 110.

There are four user-defined functions beginning at line 20. FNR takes any number and properly rounds it off to one decimal place. FNK2F ("K to F") takes a temperature in degrees Kelvin and converts it to degrees Fahrenheit. FNF2K converts degrees Fahrenheit to degrees Kelvin. FNT is the basic thermistor formula which converts resistance into temperature in degrees Kelvin.

Line 120 converts the calculated temperature to degrees Fahrenheit, rounds it off, and displays it. Line 130 converts degrees Kelvin to degrees Celsius (TC = TK − 273), rounds it off, and displays it as well. The Kelvin temperature scale is sometimes called the "absolute" temperature scale since 0 degrees Kelvin is absolute zero—the lowest possible temperature.

ON YOUR OWN

Even though the temperature is displayed to the nearest tenth of a degree, the thermistor and the A/D converter in the computer do not really provide that resolution or accuracy. You will notice that at higher temperatures such as 130 degrees F (holding the thermistor near a light bulb), the readings jump by as much as 20 degrees at a time. This is because a difference of one in the A/D reading corresponds to 20 degrees in this temperature range. The resolution is even worse at higher temperatures, but it is still adequate for many applications.

You can write a program to monitor the temperature every fifteen minutes and to graph the results. You might have the computer generate an alarm sound whenever the temperature exceeds a specified range. Data logging and remote sensing are just two of the advantages of using your computer as a thermometer.

A second thermistor can be easily added to Control Port 1. Add another wire to pin 7 (5 volts) and a wire to pin 5 (POT Y). In *Thermal Connection* read the second thermistor with PEEK(252) instead of PEEK(252) for the Commodore 64, or use POT(2) for the Commodore 128. It is possible to connect and read as many as four thermistors at once, but C-64 owners should use the four-paddle program given in the *Programmer's Reference Guide* if more than two are used.

You can measure relative humidity with two thermistors. Wrap one in a cotton wick and keep it wet. Spin it rapidly overhead in a medium-sized circle and record the temperature. Leave the other one dry and record its temperature. The temperature difference between the two is because of the evaporation rate from the wet one, and that depends upon the relative humidity. A book on meteorology or psychrometry will show you how to convert the temperature difference into relative humidity.

Let me know what other applications you come up with. Hmmm. I wonder how cold the rock salt and ice makes my ice cream freezer. I wonder if my old 1541 disk drive is really hot enough to melt steel. I wonder how well my attic insulation works. I wonder how much the temperature in my refrigerator varies throughout the day. How did I ever manage without a computer-based thermometer? □

SEE PROGRAM LISTINGS ON PAGE 66

WINDOW DRESSING

For the C-128

By Richard Curcio

A well-placed window can give a C-128 BASIC program a very sophisticated look. The WINDOW statement in BASIC 7.0, however, is essentially a no-frills command. *Window Dressing* provides four routines to enhance your text screen displays. Program 1 POKEs the machine language for *Window Dressing* into location 4864. It can be located elsewhere by changing the variable SA in line 110. The program uses 247 bytes plus 200 bytes for storage immediately after the ML. The four routines are accessed with SYS statements. If SA is the start address, then FRAME=SA, AT=SA+3, CR=SA+6, and ED=SA+9.

WINDOW FRAME

Calling FRAME quickly prints a neat box around the perimeter of the current window and (optionally) a heading at the top of the box. A window is then opened inside the box. Your WINDOW statement should therefore open a window two columns wider and two rows taller than needed. The syntax is SYS FRAME [,,,,, heading]. The five commas *must* be present if a heading is called for. The routine uses the current character color and mode (normal or reverse) for the frame. The characters used are COMMODORE A, SHIFT *, and COMMODORE S for the top, SHIFT–, cursor right, and SHIFT – for the sides, and COMMODORE Z, SHIFT * and COMMODORE X for the bottom. These characters were chosen because they appear the same in uppercase/graphics or upper/lower case. They can be changed.

The heading can be anything PRINTable: string or numeric variables or literals, color changes, cursor controls, etc. The heading begins at the upper left corner of the frame. Start the heading with a cursor right if you don't want to overwrite the corner character. There is no error checking of the length of the heading versus the width of the window. If the heading is a string variable, the LEN and RWINDOW(1) functions can be used to determine if the heading is too long for the window's width. Note that RWINDOW(0) and (1) return the number of rows or columns minus one.

Once the inner window is opened, the routine performs a "dummy" PRINT. This turns off reverse printing if it was enabled.

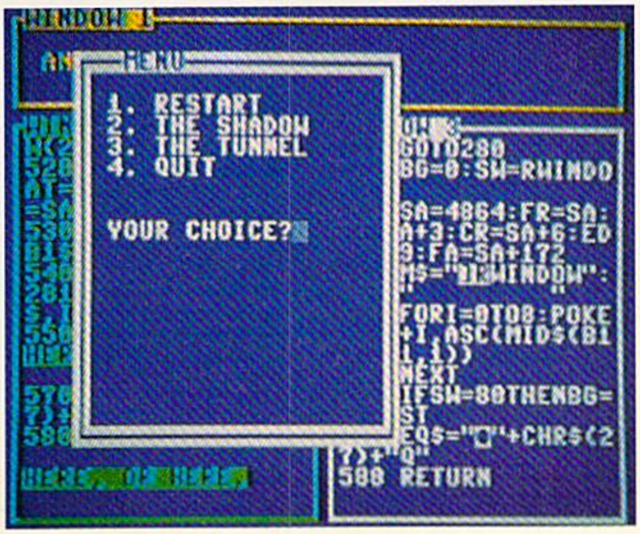
PRINT AT AND CURSOR RESTORE

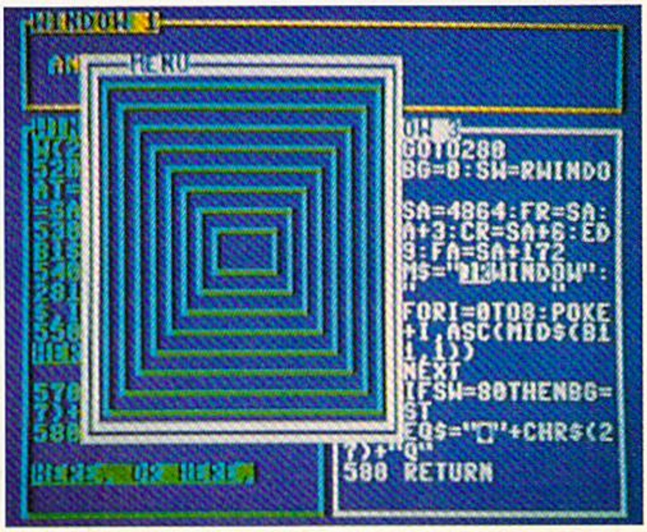
While CHAR can be used as a form of PRINTAT on a text screen, there are a few problems with this. The CHAR statement will only print characters within quotes or string variables. Numeric values must first be converted to strings using STR$. Strings must be concatendated if you want to include more than one in a CHAR statement. CHAR0,5,10,M$;H$ causes a SYNTAX error. Once CHAR has moved the cursor, it cannot easily be returned to where it came from. Early versions of the C-128 ROMs have a bug when CHAR is used in 80 columns.

The Kernal PLOT routine at 65520 (or 49176) could be used to move the cursor to a selected row and column before a PRINT statement. The "AT" routine provides a few enhancements to this approach:

```
SYS AT, flag, row, column [,,string]
```

The first value, flag, determines whether the cursor will be returned to where it was before SYS AT. This parameter cannot be omitted. If 0, the cursor is restored. Any value from 1 to 255 defers cursor restoration. The cursor position is saved, but will not be restored until SYS CR. This allows us to follow SYS AT with multiple PRINT statements before returning the cursor to its original position, if at all. Row and column refer to the current window dimensions. Note that these are in a different order than that used by





PHOTOS: MICHAEL R. DAVILA

CHAR. If row or column are beyond the dimensions of the current window, AT returns with ILLEGAL QUANTITY. Error messages for this routine and the next can be turned off by POKEing any non-zero value into SA + 21. Your program could use RWINDOW(0) or (1) to determine the window dimensions and make corrections if an upcoming row or column will be out of range.

The double commas preceding "string" must be present. String is anything PRINTable, including ESC codes. Although there are routines in ROM to save and restore the cursor position, these are used by a number of ESC characters. If AT were to use these routines, certain ESC codes in the AT string would destroy the previous cursor position. The AT routine stores the cursor column and row in more secure locations (SA+22 and 23). SYS CR restores the cursor to where it was before the most recent SYS AT.

ED

A number of locations in zero-page and page three keep track of the screen dimensions, character color, cursor location, where the tab positions are, and which screen lines are linked or continued from the previous line. These values are called the Screen Editor variables. The final routine saves and recalls five sets of editor values:

```
SYS ED, set, dir
```

where "set" is 0 - 3 and "dir" is 0 to save, and any non-zero value to recall. This will allow a program to jump from window to window, printing menus, receiving INPUT, or resuming PRINT where it left off. For four sets of editor values, the routine uses 200 bytes immediately following the ML. Storage can be moved elsewhere. Note that the contents of the screen are not saved – only those parameters mentioned above.

THE DEMO AND ALTERATIONS

Program 2 demonstrates usage of the *Window Dressing* routines. It assumes that the ML is located at address 4864. Change the value of SA in line 520 if *Window Dressing* is located elsewhere. The program determines which screen is in effect using RWINDOW(2) in line 510, and adjusts itself to 40 or 80 columns. In line 530, alternative frame characters are defined. The codes for these characters are POKEd into the cassette buffer. When needed, the FRAME routine is altered to use these characters by POKEing SA + 172 and 173 with the low byte and high byte of the first address of the new characters. (POINTER cannot be used for this because the characters must be in RAM 0 with the

Source Code for *Window Dressing*

Compiled in 64 mode using the *PAL* assembler (Pro-Line, Inc.)

```
10 SYS700
20 .OPT P,00
30 *= $1300
40 ;
50 ;-------- WINDOW DRESSING
60 ;
70 CHRGOT = $0386
80 ;
90 ;ENTRY POINTS
100 ;
110 CLV:BVC BEGIN;DRAW WINDOW FRAME
120 ;
130 CLV:BVC PRTAT;PRINT AT
140 ;
150 CLV:BVC RSTCRS;RESTORE CURSOR
160 ;
170 CLV:BVC LINKS;SAVE/RECALL ED VALS
180 ;
190 FRAME  .ASC "[c A][s *][c S][s -][RI
GHT][s -][c Z][s *][c X]";FRAME CHRS
200 ERRENB .BYT 0;ERR MSG ENABLED
210 TEMP   .BYT 0,0;HOLDS CRSR ROW/COL
220 ADDTBL .BYT 0,40,80,120,160;USED TO
CALCULATE STORAGE LOCATION
230 ;
240 BEGIN JSR $C150;CRSR HOME
250 LDA $F8
260 STA $CF;SAVE SCROLL FLAG
270 JSR $CAE5;DISABLE SCROLL
280 LDX #$FF
290 JSR FRPRT
300 LDY $E5;GET TOP ROW
310 INY
320 STY $CE
330 LFTEDG JSR FRPRT;START AT LEFT EDGE
340 INC $CE
350 LDY $CE
360 CPY $E4;HAVE WE REACHED BOTTOM"?"
370 BEQ BOTTOM
380 DEX
390 DEX
400 DEX
410 BNE LFTEDG
420 BOTTOM JSR FRPRT
430 JSR $C854;CRSR RT MOVES TO HOME
440 JSR CHKSTR;PRINT ANY HEADING
450 LDA $CF
460 STA $F8;RESTORE SCROLL FLAG
470 ;
480 SMALL INC $E5
490 INC $E6;OPEN A WINDOW
500 DEC $E4
510 DEC $E7;INSIDE THE FRAME
520 JSR $CA32;CLR SCREEN LINKS
530 JMP $C76F;PRINT RETURN & RTS
540 ;
550 PRTAT PHA;RESTORE CURSOR IF A=0
560 LDA $EC
570 STA TEMP;SAVE CRSR POS
580 LDA $EB
590 STA TEMP+1
600 MOVEIT CLC;WILL MOVE CRSR
610 JSR $C018;CALL PLOT. X=ROW, Y=COL
620 BCS ERR1;IF X & Y INVALID
630 JSR CHKSTR
640 PLA;GET ACCUML
650 BEQ RSTCRS
660 DONEPLOT RTS
670 ERR1 PLA
680 ERRMSG LDA ERRENB;IF ZERO PRINT MESS
AGE
690 BNE DONEPLOT
700 JMP $7D28;ILLQTY
710 ;
720 LINKS CMP #$05
730 BCC LINK2
740 BCS ERRMSG
750 ;
760 RSTCRS LDA TEMP
770 STA $EC
780 LDA TEMP+1
790 STA $EB
800 JMP $C15C;SET POINTERS
810 ;
820 CHKSTR JSR CHRGOT;PRINT ANYTHING"?"
830 BEQ DONEPLOT;NO
840 JSR $795C;CHK COMMA
850 JMP $555A;PRT STRING
860 ;
870 FRPRT JSR PRTIT;LEFT EDGE
880 LDY $E6;LEFT COL.
890 INY
900 INX
910 CENT JSR PRTIT+1;CENTER
920 INY
930 CPY $E7;RIGHT COL
940 BNE CENT
950 ;
960 PRTIT INX
970 LDA FRAME,X
980 JMP $C00C;PRINT CHR IN A & RTS
990 ;
1000 LINK2 STX $C3
1010 TAY
1020 LDA #<AREA
1030 LDX #>AREA
1040 CLC
1050 ADC ADDTBL,Y
1060 BCC LINK3
1070 INX
1080 LINK3 LDY $C3
1090 STA $C3
1100 STX $C4
1110 LDX #$1A
1120 TYA
1130 BNE LINK6
1140 LDY #$25
1150 LINK4 DEY
1160 LDA $E0,X
1170 STA ($C3),Y
1180 DEX
1190 BPL LINK4
1200 LDX #$0D
1210 LINK5 DEY
1220 LDA $0354,X
1230 STA ($C3),Y
1240 DEX
1250 BPL LINK5
1260 RTS
1270 ;
1280 LINK6 LDY #$25
1290 LINK7 DEY
1300 LDA ($C3),Y
1310 STA $E0,X
1320 DEX
1330 BPL LINK7
1340 LDX #$0D
1350 LINK8 DEY
1360 LDA ($C3),Y
1370 STA $0354,X
1380 DEX
1390 BPL LINK8
1400 RTS
1410 ;
1420 AREA .BYT 0;EDITOR STORAGE BEGINS
```

PRESERVING 80 COLUMN SCREENS

The memory-moving feature of *TextSave* (January 1988) can be combined with SYS ED to save and recall two 80 column screens. In this way, you could save the screen, open a window, and then recall the screen, replacing what was overwritten by the window.

First, some memory must be set aside in RAM 0 to hold the 4K of each 80 column screen:

```
GRAPHIC 1: GRAPHIC 0
```

sets aside 9K of memory beginning to address 7168 normally used for a bit map display. This area will be safe until a GRAPHIC CLR command. The BASIC subroutine to save/recall 80 columns should look something like this:

```
2000 POKE 195,0: POKE 196,SH: POKE 174
,0: POKE 175,EH
2010 SYS TS+236,0,0,D1: SYS ED,S,D2: R
ETURN
```

Line 2000 sets up the start and end address (+1) in RAM 0 for *TextSave* to move. "SH" is the start address divided by 256 and "EH" is 4096 bytes higher, also divided by 256. In line 2010, "TS" is the start address of *TextSave* and 0, 0, tells the routine to start at address 0 in VDC memory, the normal location of 80 column text. Direction D1 is 0 to move data to 80 column memory and D2 is > 0 to restore screen editor values. To save an 80 column screen, D1 is > 0 and D2 = 0. "S" is the set of editor values. It's up to you to keep track of which set goes with which display.

ML.) The default characters can also be changed by POKEing new codes into SA + 12 through SA + 20 in the order described earlier.

In line 340, SYS 52591 calls a ROM routine to turn on the cursor. This provides a cursor for use with GET or GETKEY as an alternative to INPUT. The cursor is turned off by calling another ROM routine at location 52639 with 32 in the accumulator.

To change the location of editor storage, POKE the low byte and high byte of the storage address into SA + 181 and SA + 183. Each set requires 40 bytes. If error messages are disabled as described earlier, storage can be in RAM 0 above 16383. The computer must be in the BANK 0 configuration in this case, before SYS ED. For the other routines, the C-128 must be in BANK 15 since several ROM routines are used. "AT" and "ED" set the processor carry flag if any values are out of range, so RREG,,,SR, immediately after the SYS, reads the status register into SR. IF SR AND 1 = 1 THEN something went wrong.

If error messages are enabled, the TRAP statement of BASIC 7.0 can be used to prevent ILLEGAL QUANTITY from messing up your screen. Note that parameters greater than 255, or strings where numbers should be, cause SYS to return with the appropriate error message regardless of the condition of the *Window Dressing* error message flag. ☐ ***SEE PROGRAM LISTING ON PAGE 69***

FLOTSAM

Send your comments on any aspect of Commodore computing to *Flotsam*, c/o *Ahoy!*, Ion International Inc., 45 West 34th Street—Suite 500, New York, NY 10001.

I would like to thank you for the past few years of very good reading. The magazine is well-balanced, with a wide variety of articles. It seems to cover all the different aspects of Commodore computing, not to mention the talented writers. Of course, I do have a complaint or two. I don't feel that the average Commodore user is that interested in COMAL. This section could possibly be used to teach beginners BASIC, ML, everyday commands, etc.

I am an average user who is eager to learn in a non-complicated way (like most people I talk to). Between work and life's other necessities, I don't have the time to study as much as I should, so I like your magazine and its quick tips. Again, thank you for making computing on my C-128 less frustrating and more enjoyable.
—John Burke
Rosedale, NY

We began our COMAL Column *because we felt the language had not received the press it deserved. But space, as you indicate, is at a premium. As of the next installment (September), COMAL coverage will cease for the time being.*

There is very little software available for the C-128 in its 128 mode, and there appears to be little on the horizon. Therefore, when *Ahoy!* publishes a program for the 128 that uses some of the great characteristics of the machine, 128 owners are naturally very enthusiastic.

I recently read that Commodore had sold one million C-128's. That is a large number for potential sales of software and is generally being ignored by the software houses. However, there is a possible alternative. If *Ahoy!* were to publish a special issue on a yearly basis dedicated only to the C-128, I believe that C-128 owners would feel that they had died and gone to heaven and would make such a venture a huge success. With Blakemore, Rupert, et al, you certainly have the talent to put it over.
—Neal Smith
Rochester, NY

We'd love to do as you suggest, Neal. But remember the Aesop's fable about the man with the donkey? See below.

I first became interested in your mag because of *Tips Ahoy!* As a relatively new Commodore owner, I was hungry for programming hints and tricks. I was satisfied for awhile, but it seems lately you've been allotting an unproportionate amount of space to the 128 user. Please don't assume that just because breakthroughs in computer technology have been advancing at phenomenal rates that all of us 64 owners have been experiencing the same phenomenon with regards to our salaries. Heck, there are a lot of us that haven't even utilized the potential of our 64's yet. I don't think your readers would mind a couple fewer game programs if it meant expanding your more useful columns (i.e., *Tips Ahoy!* and *Flotsam*).
—Tom Rowan
Ft. Wayne, IN

TIPS AHOY!

Send your best programming and hardware hints for the C-64, C-128, or Amiga to *Tips Ahoy!*, c/o Ion International Inc., 45 W. 34th St.–Suite 500, New York, NY 10001. Include a stamped and self-addressed envelope.

LET YOUR FUNCTION KEYS ESCAPE

The Commodore 128 has numerous useful and entertaining ESC-key combinations, some of which are so darn good you wonder why Commodore didn't put a specific key for them right on the keyboard. ESC-D, for instance: that one deletes one line on the screen, moving all the text after it up one line. I use it all the time! But I'm lazy, and absolutely hate having to press two keys to do anything. So in the interest of cutting my typing in half:

```
KEY7,CHR$(27)+"D"
```

Now when F7 gets pressed, it does the same thing as an ESC-D. Follow that up with:

```
KEY8,CHR$(27)+"I"
```

and pressing F8 inserts a line. I know, I know, we're back up to two keypresses, SHIFT plus F7, but now they're in the same corner of the keyboard. (And easier to find at 3:00 a.m.!) After all, they're related functions, right? Now try these:

```
KEY1,CHR$(27)+"A"+CHR$(27)+"E" :REM AUTO
INSERT ON, CURSOR FLASH OFF
KEY2,CHR$(27)+"C"+CHR$(27)+"F" :REM AUTO
INSERT OFF, CURSOR FLASH ON
KEY3,CHR$(27)+"V" :REM SCROLL TEXT UP ON
E LINE
KEY4,CHR$(27)+"W" :REM SCROLL TEXT DOWN
ONE LINE
```

Nothing fancy, just some more "related" ESC codes and suggested function keys to store them in. If you really like a particular sequence, you can preserve it forever by typing a line number before the KEY command:

```
1000 COLOR0,1:COLOR4,12
1010 KEY5,CHR$(27)+"T" :REM SET UPPER LE
FT CORNER OF WINDOW
1020 KEY6,CHR$(27)+"B" :REM SET LOWER RI
GHT CORNER OF WINDOW
```

I have a more elaborate version of that little program saved on my main C-128 programming disk, named "KEYS". The first thing I do when I turn on my 128 is type RUN"KEYS" and all my favorite function key combos are restored, along with a pleasing screen color. You can build tremendously elaborate ESC-key combos into the function keys.

–David F. Paulsen
Houston, TX

Compiled by Michael R. Davila

RIPPLES

This short program for the C-64 and C-128 plots several concentric circles in extended background mode, then cycles through the different registers to give the illusion of motion. It takes about five minutes to plot all the circles, so be patient. This is one of those deals where the display looks good, but it's left up to you to find a good application.

–Cleveland M. Blakemore

```
·10 REM RIPPLES FOR THE C-64 OR C-128
·20 V=53248:N=6.3:H=20:W=12:M=19:U=1:L=.5
 :K=40:B=1024:J=24:G=64:S=32:OF=54272
·30 PRINT"[CLEAR]":POKE53281,.:POKE53280,
 .:POKE53265,PEEK(53265)OR64
·40 POKEV+34,1:POKEV+35,12:POKEV+36,11
·50 E=30/25:C=1:FORR=MTOUSTEP-U
·60 FORT=.TONSTEP(U/(R+R)):X=INT(E*R*COS(
 T)+H+L):Y=INT(R*SIN(T)+W+L)
·70 P=Y*K+X+B:IFX>.ANDX<KANDY>.ANDY<JTHEN
 POKEP,S+G*C
·80 NEXT:C=C+U+(C=3)*3:NEXT
·90 POKEV+34,0:POKEV+35,0:POKEV+36,6
·100 A=PEEK(V+34):B=PEEK(V+35):C=PEEK(V+3
 6):POKEV+34,B:POKEV+35,C:POKEV+36,A
·110 FORX=.TO60:NEXT:GOTO100
```

SPLIT FACE

Split Face scrolls half the screen left and the other half right. All character colors are maintained as your screen takes on a split personality. Now you have a clear screen that's ready for your next title, menu, or graphics display.

Append *Split Face* to your own programs and have them run the loader to POKE the data into memory. It works on the C-64 and C-128 in 40 column mode. Whenever you want *Split Face* to drop the hatchet, enter the following:

```
100 SYS 49152
```

For the C-128 it would be:

```
100 SYS 4864
```

The line numbers are only examples. You can use any line number you want.

To demonstrate, run the loader. Now list the program to fill the screen. Cursor to the top and enter in direct mode (no line number):

```
SYS 49152
```

(SYS 4864 for the C-128), then press RETURN. Your characters will split like a knife through butter.

Split Face can be relocated by changing the variable A

Lyco Computer
Marketing & Consultants

Since 1981

Lyco Means Total Service.



Mark "Mac" Bowser, Sales Manager

I would personally like to thank all of our past customers for helping to make Lyco Computer one of the largest mail order companies and a leader in the industry. Also, I would like to extend my personal invitation to all computer enthusiasts who have not experienced the services that we provide. Please call our trained sales staff at our toll-free number to inquire about our diverse product line and weekly specials.

First and foremost our philosophy is to keep abreast of the changing market so that we can provide you with not only factory-fresh merchandise but also the newest models offered by the manufacturers at the absolute best possible prices. We offer the widest selection of computer hardware, software and accessories.

Feel free to call Lyco if you want to know more about a particular item. I can't stress enough that our toll-free number is not just for orders. Many companies have a toll-free number for ordering, but if you just want to ask a question about a product, you have to make a toll call. Not at Lyco. Our trained sales staff is knowledgeable about all the products we stock and is happy to answer any questions you may have. We will do our best to make sure that the product you select will fit your application. We also have Saturday hours — one more reason to call us for all your computer needs.

Once you've placed your order with Lyco, we don't forget about you. Our friendly, professional customer service representatives will find answers to your questions about the status of an order, warranties, product availability, or prices.

Lyco Computer stocks a multimillion dollar inventory of factory-fresh merchandise. Chances are we have exactly what you want right in our warehouse. And that means you'll get it fast. In fact, orders are normally shipped within 24 hours. Free shipping on prepaid orders over $50, and there is no deposit required on C.O.D. orders. Air freight or UPS Blue/Red Label shipping is available, too. And all products carry the full manufacturers' warranties.

I can't see why anyone would shop anywhere else. Selection from our huge in-stock inventory, best price, service that can't be beat—we've got it all here at Lyco Computer.

TO ORDER, CALL TOLL-FREE: 1-800-233-8760
New PA Wats: 1-800-233-8760
Outside Continental US Call: **1-717-494-1030**

Hours: 9AM to 8PM, Mon. - Thurs.
9AM to 6PM, Friday — 10AM to 6PM, Saturday

For Customer Service, call 1-717-494-1670,
9AM to 5PM, Mon. - Fri.
Or write: Lyco Computer, Inc.
P.O. Box 5088, Jersey Shore, PA 17740

VISA® MasterCard

C.O.D. Risk-Free Policy: • full manufacturers' warranties • no sales tax outside PA • prices show 4% cash discount; add 4% for credit cards • APO, FPO, international: add $5 plus 3% for priority • 4-week clearance on personal checks • we check for credit card theft • sorry, compatibility not guaranteed • return authorization required • due to new product guarantee, return restrictions apply • price/availability subject to change • prepaid orders under $50 in Continental US, add $3.00

Monitors

Thomson:

230 Amber TTL/12"	$79.95
4120 CGA	$219.95
4160 CGA	$254.95
4460 EGA	$349.95
GB 200 Super Card	$184.95
4570	$CALL

THOMSON
4120 Monitor $225.95

Magnavox:

BM7652	$79.95
BM7622	$79.95
7BM-613	$75.00
7BM-623	$79.95
CM8502	$179.95
CM8505	$199.95
9CM-053	$339.95
CM8762	$239.95
8CM-515	$249.95
CM9043	$329.95

Blue Chip:

BCM 12" Green TTL	$64.95
BCM 12" Amber TTL	$69.95

NEC:

Multisync II	$599.95

Modems

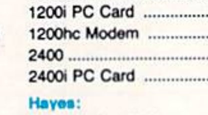
Avatex:

1200e	$69.95
1200i PC Card	$69.95
1200hc Modem	$89.95
2400	$179.95
2400i PC Card	$169.95

Hayes:

Smartmodem 300	$149.95
Smartmodem 1200	$285.95
Smartmodem 2400	$425.95

Smarteam
1200 Baud Modem



OUR PRICE $89.95

Hayes Compatible

COMMODORE

Activision:

Champion. Basketball	$19.95
Music Studio	$19.95
Leather Goddesses	$22.95
Top Fuel Eliminator	$15.95
Beyond Zork	$25.95
GFL Football	$19.95
Gee Bee Air Rally	$16.95
Last Ninja	$19.95
Might & Majic	$19.95
Nord & Bert	$19.95
Aliens	$25.95
Maniac Mansion	$19.95

Batteries Included:

Paperclip III	$31.95
Outrageous Pages	$31.95

Berkeley Softworks:

Geofile C64	$29.95
Geo Calc C64	$29.95
Geos 64	$35.95
Geos 128	$39.95
Geowrite	$29.95
Geowrite 128	$39.95
Geopublish C64	$39.95
Berkeley TriPak	$29.95

Broderbund:

Bank St. Writer	$27.95
Carmen San Diego	$19.95
Graphic Lib. I, II, III	$13.95
Print Shop	$25.49
Print Shop Compan.	$22.95
Cauldron	$16.95
Superbike Challenge	$11.95
Magnetron	$14.95

Electronic Arts:

Marble Madness	$20.95
Yeager's AFT	$22.95
Demon Stalkers	$20.95
Dragon's Lair	$16.95
Skate or Die	$20.95
Strike Fleet	$20.95

Epyx:

Destroyer	$22.95
Fastload	$19.95
Sub Battle	$22.95
Winter Games	$11.95
California Games	$22.95
Str. Sports Basketball	$22.95
Summer Games II	$11.95
World Games	$22.95
Boulderdash Con Set	$13.95
Rad Warrior	$13.95

Firebird:

Elite	$18.95
Guild of Thieves	$22.95
Pawn	$22.95
Tracker	$22.95

COMMODORE

Strategic Simulations:

Gettysburg	$33.95
Phantasie II	$22.95
Phantasie III	$22.95
Road War 2000	$22.95
Wizards Crown	$22.95
Wargame Constr.	$16.95
Battlecruiser	$33.95
Eternal Dagger	$22.95
Shiloh	$22.95
Questron II	$22.95
Phantasie	$22.95
Sons of Liberty	$19.95

Sublogic:

Flight Simulator II	$31.49
Jet Simulator	$24.95
Night Mission Pinball	$18.95
Scenery Disk 1-6	$12.95
Stealth Mission	$31.95

Timeworks:

Partner C64	$22.95
Partner 128	$27.95
Swift Calc 128	$27.95

AMIGA

Access:

World Class Lead. Bd.	$CALL

Activision:

Champ. Basketball	$25.95
Championship Golf	$22.95
GFL Football	$25.95
Gee Bee Air Rally	$22.95

Electronic Arts:

Gridiron	$26.95
Oneon One	$13.95
Weaver Baseball	$33.95
Return to Atlantis	$CALL

Epyx:

Apshai Trilogy	$11.95
Winter Games	$22.95
World Games	$22.95
Destroyer	$22.95

Firebird:

Guild of Thieves	$25.95
Pawn	$25.95
Starglider	$25.95

Microprose:

Silent Service	$22.95

Sublogic:

Flight Simulator II	$31.49
Scenery Disk	$CALL

Unison World:

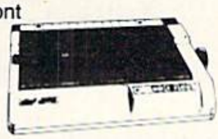
Print Master	$25.95
Art Gallery 1 or 2	$14.95

1-800-233-8760

New for '88 — Commodore Ready — Great Price

star micronics·inc NX-1000

- 144 cps Draft
- 36 cps NLQ
- EZ Operation Front Panel Control



$179.95

SEIKOSHA SP 180VC

$129.95



- 100 cps draft
- 20 cps NLQ

Quantities Limited

Panasonic 1091i *Model II*



- 192 cps Draft
- NLQ Mode
- Excellent Versatility

$199.95

PRINTERS

star micronics·inc

NX-1000	$179.95
NX-1000C	$179.95
NX-1000 Color	$225.95
NX-1000C Color	$229.95
NX-15	$309.95
NR-10	$339.95
NR-15	$439.95
NB-15 24 Pin	$699.95
NB24-10 24 Pin	$425.95
NB24-15 24 Pin	$579.95

SEIKOSHA

SP 180Ai	$129.95
SP 180VC	$129.95
SP 1000VC	$139.95
SP 1000AP	$169.95
SP 1200VC	$155.95
SP 1200Ai	$165.95
SP 1200AS RS232	$165.95
SL 80Ai	$299.95
MP1300Ai	$269.95
MP5300Ai	$399.95
MP5420 Fa	$1079.95
SP Series Ribbon	$7.95
SK3000 Ai	$339.95
SK3005 Ai	$419.95
SPB 10	$CALL
SL 130Ai	$599.95

EPSON

LX800	$199.95
FX86E	$279.95
FX286E	$424.95
EX800	$399.95
LQ500	$339.95
LQ1000 w/Tractor	$549.95
LQ2500	$819.95
GQ3500	$LOW
LQ850	$525.95
LQ1050	$715.95

Panasonic

1080i Model II	$179.95
1091i Model II	$199.95
1092i	$319.95
1592	$409.95
1595	$459.95
3131	$299.95
3151	$479.95
KXP 4450 Laser	$CALL
1524 24 Pin	$559.95
Fax Partner	$589.95

CITIZEN

120 D	$169.95
180 D	$189.95
MSP-10	$259.95
MSP-40	$309.95
MSP-15	$349.95
MSP-50	$399.95
MSP-45	$459.95
MSP-55	$539.95
Premiere 35	$499.95
Tribute 224	$649.95

DIABLO

D25	$499.95
635	$779.95

OKIDATA

Okimate 20	$119
Okimate 20 w/cart	$179.95
120	$189.95
180	$219.95
182	$209.95
182+	$225.95
183	$249.95
192+	$339.95
193+	$449.95
292 w/interface	$449.95
293 w/interface	$585.95
294 w/interface	$819.95
393	$955.95

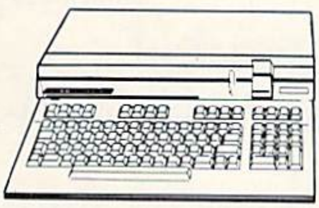
BROTHER

M1109	$164.95
M1409	$299.95
M1509	$335.95
M1709	$475.95
Twinwriter 6 Dot & Daisy	$899.95
M1724L	$619.95
HR20	$345.95
HR40	$599.95
HR60	$709.95

Toshiba

321SL	$489
341 SL	$659
P351 Model II	$899
351 SX 400 cps	$1019

COMMODORE 128D System



$449.95

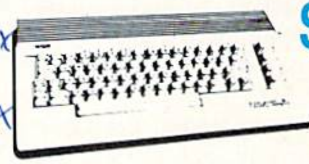
Commodore 128D Plus Thomson 4120 Monitor $649.95

COMMODORE HARDWARE

1541 II Disk Drive	$175.95
1581 Disk Drive	$189.95
Indus GT C-64 Drive	$169.95
1802C Monitor	$189.95
64 C Computer	$169.95
128D Computer/Drive	$449.95
C-1351 Mouse	$39.95
1700 RAM	$109.95
C-1750 RAM	$CALL
1764 RAM C64	$117.95
1084 Monitor	$299.95
128 Computer	$219.95
Excel 2001 C128 Drive	$199.95
Excel FSD-2+ C64 Drive	$149.95

COMMODORE 64C COMPUTER

$169.95



Color System HEADSTART by Vendex

- Ready to plug in and use immediately
- Ultra fast 8 Mhz Intel 8088-2 Processor
- 512K RAM memory expandable to 768K
- 2-360K disk drives standard
- First complete system with clock calendar and built-in ports for printer, RS232, 2 joysticks, mouse and light pen
- Includes $500 worth FREE software programs
- Hi Res color monitor included!



$1049.95

PC COMPATIBLE HARDWARE

BCM 12G Gr. Monitor	$64.95
BCM 12A Am. Monitor	$69.95
BCC CG Color Card	$94.99
AST	$CALL
Laser Compact XT	$475.95
Laser Compact XTE 640K	$549.95
Laser Desktop Turbo XT 256K	$569.95
Laser Desktop Turbo XT 640K	$599.95

PERSONAL COMPUTER



- 4.77 Mhz speed
- Dual floppy
- Green monitor included

$579.95

Attention Educational Institutions: If you are not currently using our educational service program, please call our representatives for details.

Reader Service No. 157

in line 1. Be sure to SYS to the new address if you do change it.

—Buck Childress
Salem, OR

```
•1 A=49152:REM * (A=4864 FOR THE C-128) *
  (BOTH VERSIONS CAN BE RELOCATED.) *
•2 FORJ=ATOA+107:READB:POKEJ,B:X=X+B:NEXT
 J
•3 IFX<>17061THENPRINT"ERROR IN DATA[3"."
 ]":END
•4 REM *** YOU COULD PLACE A RETURN HERE
 IF THIS IS USED AS A SUBROUTINE. ***
•5 DATA169,20,133,250,169,0,162,4,160,216
 ,133,251
•6 DATA133,253,134,252,132,254,162,24,160
 ,38,177,251
•7 DATA200,145,251,136,177,253,200,145,25
 3,136,136,192
•8 DATA20,176,239,160,20,169,32,145,251,1
 60,1,177
•9 DATA251,136,145,251,200,177,253,136,14
 5,253,200,200
•10 DATA192,21,144,239,160,19,169,32,145,
 251,165,251
•11 DATA24,105,40,133,251,144,2,230,252,1
 65,253,24
•12 DATA105,40,133,253,144,2,230,254,202,
 16,181,160
•13 DATA10,202,208,253,136,208,250,198,25
 0,208,153,96
```

SURVEYOR

1000 screen and 1000 color RAM locations! Now just how are we as mere mortals supposed to look at the monitor and know exactly where each one is? If you're trying to POKE characters and colors to various locations, it can quickly become hair-pulling time.

You can always use the chance method. Throw darts at the monitor from 10 paces and figure that's close enough. Trouble is, they don't stick too well. And, if they do, you'll wind up with an air-conditioned screen.

Take heart. There is a better way. And it's 100% accurate. *Surveyor* instantly gives you the screen and color RAM locations of any place you choose. Let's give it a whirl.

First load and run *Surveyor.* To activate it, type SYS 49152 and press RETURN. Now you can survey the situation.

Just cursor to a desired target and press the SHIFT and CONTROL keys at the same time. Two numbers will appear. The first is the screen RAM location and the second is the color RAM location. Try it again. Nice and easy, nice and quick.

Surveyor can easily be relocated if you want. Just change the variable A in line 2 to your new starting address. Don't forget to SYS there if you do.

RUN STOP/RESTORE deactivates *Surveyor.* SYS to the appropriate address to reactivate it.

Don't pull your hair out. But, if you do, maybe you can sell it for a toupee.

—Buck Childress
Salem, OR

```
•1 REM *** SURVEYOR C-64 ***
•2 A=49152:PRINTCHR$(147)
•3 FORJ=ATOA+95:READB:POKEJ,B:X=X+B:NEXTJ
•4 IFX<>14696THENPRINT"ERROR IN DATA[3"."
 ]":END
•5 C=INT(A/256):D=(A-(C*256))+13:POKEA+1,
 D:POKEA+3,C
•6 PRINT"DATA OK * SYS"A"TO ACTIVATE[3"."
 ]":END
•7 DATA169,13,162,192,120,141,143,2,142,1
 44,2,88
•8 DATA96,173,141,2,166,254,133,254,201,5
 ,208,69
•9 DATA228,254,240,65,169,1,133,204,165,2
 11,133,252
•10 DATA165,214,133,253,165,209,166,210,2
 4,101,252,144
•11 DATA1,232,134,251,170,165,251,32,205,
 189,169,42
•12 DATA32,210,255,165,243,166,244,24,101
 ,252,144,1
•13 DATA232,134,251,170,165,251,32,205,18
 9,166,253,164
•14 DATA252,24,32,240,255,169,0,133,207,7
 6,72,235
```

FASTER CHARACTERS IN BASIC 7.0

The following program redefines the character set in only two seconds. A FOR/NEXT loop would take 32 seconds.

Lines 50 and 60 move the character ROM into the bit map screen. The monitor is called via a function key.

Lines 80 through 110 give you underlined letters. To access them, press the letter keys with the logo key. The effect is much better if you use capitals in line 80.

The next 12 lines are for simple animation. Circles are drawn on one character then copied to the space character.

—James Glaser
Burney, CA

```
•10 COLOR 0,1:COLOR 1,4:COLOR 4,1
•20 GRAPHIC 1,1:GRAPHIC 0,1:BANK 15
•30 PRINT "[HOME][HOME][3"[DOWN]"]"TAB(15
 )"WORKING[3"."]"
•40 WINDOW 0,13,39,24,1
•50 KEY 1,"[CLEAR][BLACK]MONITOR"+CHR$(13
 )+"T ED800 ED[3"F"] E3800"+CHR$(13)+"X"+
 CHR$(13)+"GOTO70"+CHR$(13)
•60 POKE 209,PEEK(4096):POKE 210,0:END
•70 KEY 1,"GRAPHIC":GRAPHIC 2,0,13:COLOR
 5,14
•80 TEXT$="<->.* [s K][s I][s T]=[s G]+[s
  M]!"+CHR$(34)+"[s N][s Q][s D][s Z][s S
 ][s P][s A][s E][s R][s W][s H][s J][s L
 ][s Y][s U][s O]?[s F][s C][s X][s V][s
 B]"
•90 POKE 4588,216:CHAR,19,21,TEXT$
•100 FOR A=91 TO 127:R=14343+A*8
```

Continued on page 82

THEY-DO-RUN-RUN-RUN-THEY-DO-RUN-RUN

By Richard Herring

Speed. Raw, blinding, power-user speed. Amigas running anywhere from 7.14 megahertz to over 14 MHz with turbo boards sharpening the cutting edge.

How can the poor C-64 keep up at a piddley 1 MHz? Or even the C-128 in fast mode at a dawdling 2 MHz? But obviously they do keep up, as is evidenced by some of the fine and very powerful software available.

For a given computer, speed can be significantly enhanced by two things: programmer's tricks and programmer's tools. Tricks come with experience. Tools are equally available to all of us—often off the shelf.

One key tool is the language in which we each choose to program. Several of you have asked just how fast COMAL is. Forget for the moment how its structure enhances programming and debugging. If COMAL is not at least as fast as other languages, then it's not what some of you power programmers are looking for.

Enter Herbert Denaci, engineer and speed aficionado. While I was doing some of the background work on this column, he and I exchanged a few letters. Denaci has done exactly the kind of speed testing that will be useful in proving COMAL's value. So we'll explore his benchmarks, as well as a few others, and you can judge COMAL for yourself.

Denaci's Benchmarks—C-64 or C-128 at 1 MHz

(Scores are ranked against BASIC 2.0. A score of .50 means the benchmark ran in half the time BASIC took.)

	Sieve	Trig	Flight	Ahl's
COMAL 0.14	.74	1.00	.69	.97
COMAL 2.0	.30	.98	.54	.22
BASIC 2.0	1.00	1.00	1.00	1.00
BASIC 7.0	1.46	1.08	1.28	1.09
Pascal (Oxford)*	.14	1.05	.67	.55
PROMAL*	.06	1.13	.86	1.45

*The "run" times for Pascal and PROMAL do not include the separate step of compiling the source code. Compile times vary from 7 to 94 seconds on these benchmarks, which run in anywhere from ½ to 5 minutes.

Now no benchmark gives an absolutely true picture of a language's speed. What you want as a programmer is the most speed for the particular program you're writing right now. The useful speed of the language will change as the demands of your program vary. All a benchmark can do is provide a comparison based on the features it happens to exploit.

Let's look first at the Sieve of Eratosthenes. It has been used as a programming algorithm since at least 1969. All it does is find prime numbers. Sieve is generally used to test a high-level language's performance in Boolean algebra. It also gives an indication of the language's abilities in array handling, memory references, and structured control statements.

The heart of Denaci's translation of Sieve, into COMAL 0.14, is:

```
0130 FOR I#:=0 TO SI# DO
0140   IF FLAGS#(I#) THEN
0150     PRIME#:=I#+I#+3
0160     K#:=I#+PRIME#
0170     WHILE K#<=SI# DO
0180       FLAGS#(K#):=FALSE
0190       K#:+PRIME#
0200     ENDWHILE
0210     COUNT#:+1
0220   ENDIF
0230 ENDFOR I#
```

Sieve has a lot of calculations to perform. Writing it this way allows it to be a real screamer because: 1) it doesn't use division, 2) it uses prior knowledge about numbers that cannot be primes (even numbers), 3) it already knows that 0, 1, 2, and 3 are the first four primes, and 4) it calculates primes as the value of the variable PRIME#, or twice the current array index plus three.

To stretch out the test a little, a major loop is run 10 times with these additional lines:

```
0080 DIM FLAGS#(0:SI#)
0090 FOR ITER#:=1 TO ML# DO
0100   FOR I#:=0 TO SI# DO FLAGS#(I#):=TR
UE
0110   PRINT "LOOP ",ITER#
0120   COUNT#:=0
```

```
0240 ENDFOR ITER#
0250 PRINT "COUNT=";COUNT#
```

Denaci's Benchmarks—C-128 at 2 MHz

(Scored against BASIC 2.0)

	Sieve	Trig	Flight	Ahl's
COMAL 0.14	.73	1.03	.78	.97
COMAL 2.0	.30	1.00	.60	.22
BASIC 2.0	1.00	1.00	1.00	1.00
BASIC 7.0	1.38	1.21	1.35	1.04
FORTRAN (Nevada)*	–	5.70	2.84	1.60

*The run times for FORTRAN use the C-128's Z80 microprocessor and do not include the separate compile times.

Denaci's calls his second benchmark Trig. It uses trigonometry functions that are typical of those used in flight dynamics problems. The heart of this benchmark, in COMAL 0.14, is:

```
0100 ITERATION:=0
0110 A:=.1
0120 B:=.2
0130 FOR ITERATION:=1 TO 1000 DO
0140   C:=SIN(A/B)*COS(A/B)
0150 ENDFOR ITERATION
```

The Trig benchmark is just a warmup for Denaci's real interest—the Flight Dynamics Simulation benchmark. He wrote this 145-line benchmark to be representative of a guided missile program used for design studies. It's not listed here for space reasons—not physical space, but the space that appeared between my ears when I attempted to decipher it. Suffice it to say that Flight contains plenty of arithmetic calculations as well as all the trigonometry you could hope for.

Dropping back in time to another old benchmark, we find David Ahl's Simple benchmark from the January '84 issue of *Creative Computing*. Translated into COMAL 0.14, the meat of this test looks like:

```
0100 R:=0
0110 S:=0
0120 FOR N#:=1 TO 100 DO
0130  A:=N#
0140  FOR I#:=1 TO 10 DO
0150   A:=SQR(A)
0160   R:+RND(1)
0170  ENDFOR I#
0180  FOR I#:=1 TO 10 DO
0190   A:=A^2
0200   R:+RND(1)
0210  ENDFOR I#
0220  S:+A
0230 ENDFOR N#
```

This benchmark tests for mathematical accuracy by taking the square root of A, then squaring the result and adding it to the variable S. Perfect accuracy would result in S equaling 5050 at the end of 100 cycles through

TECH NOTES

1) The COMAL Users Group claims version 2.0 is about twice as fast as 0.14. *COMAL Today* magazine #6, p. 17.

2) All the C-128 benchmarks were run in C-64 mode. COMAL 0.14 won't load in C-128 mode and the COMAL 2.0 cartridge forces the C-128 into C-64 mode on bootup.

3) If you own a C-128, you can kick your computer into its fast speed while in the 64 mode. Normally the C-128 in 64 mode operates with a 1 MHz clock, but you can switch to the 2 MHz clock with POKE 53296,3 and reset to 1 MHz with POKE 53296,0. In the faster speed, the C-128 screen will go blank and internal computer operations will click along at twice the normal speed. (The corresponding hex address to 53296 is $D030.) You can elect to just blank (not erase) the screen and take the modest speed improvement by POKEing 53265 with a 3 (off) and then a 0 (on).

4) The Sieve benchmark has been written other ways using division. It runs dozens of times slower. The moral? If you're looking for speed, don't throw out high-level languages; look for better algorithms. High-level languages are generally considered to save time in development and debugging. If you just have to have more speed in your finished program, try rewriting the most critical routines in assembly language. Even then, don't throw out that high-level code—it may be useful documentation in the future.

5) Ahl's benchmark is, like the others, designed to give a specific comparison of processing speed. However, it also reports on the accuracy of the language and the computer at squaring the square root of a number and coming up with exactly the original number. At the end of the completed benchmark, it subtracts the original number from the one calculated by the computer. Presumably the closer to zero the better. Here are some interesting "accuracy scores":

COMAL 0.14	0.000298	**COMAL 2.0**	0.000117
BASIC 2.0	0.001041	**BASIC 7.0**	0.001052

6) No benchmark can be the sole criterion for you to use in judging a language. Sieve, for example, does not test features like recursion or sets, which are critical to some programming tasks. And benchmarks totally fail to show ease of programming. As Denaci wrote, "In comparing the five languages, there is no doubt that COMAL is superior, it is user friendly! BASIC isn't as difficult as the "compiled" languages,

the major loop. And the benchmark exercises the random (RND) function by adding a random number between 0 and 1 to the variable R a total of 2000 times. That's 10 times (line 140) plus 10 times (line 180) multiplied by 100 (line 120). The resulting value of R, based on probabilities, should be 1000, but the actual calculation doesn't mean much with the exception of testing execution speed.

Rupert's "Benchmarks"

(Scored against BASIC 2.0)

	Addition (1 MHz / 2 MHz)	Multiplication (1 MHz / 2 MHz)
COMAL 0.14	1.07 / 1.06	1.05 / 1.04
COMAL 2.0	.64 / .64	.77 / :77
BASIC 2.0	1.00 / 1.00	1.00 / 1.00
BASIC 7.0	1.42 / 1.39	1.29 / 1.26

Next we'll borrow from the *Rupert Report* in the June 1987 issue of *Ahoy!* that showed BASIC's ability to perform 1000 additions or multiplications. Dale's test, converted to COMAL 0.14, is:

```
0100 A:=3.21896543
0110 B:=1.02104539
0120 POKE 160,0
0130 POKE 161,0
0140 POKE 162,0
0150 FOR N:=1 TO 1000 DO
0160  A:=A+B //  ++ A:=A*B
0170 ENDFOR N
0180 PRINT (256*256*PEEK(160)+256*PEEK(1
61)+PEEK(162))/60,
0190 PRINT " SECONDS"
```

but it has the limitation of only recognizing the first two letters in variable names. Using compilers makes it difficult and time consuming to troubleshoot and edit a program."

7) If you choose to run these benchmarks yourself, you can get more accurate timing than is possible with a sweep-second hand or by counting one-thous-and-and-one. Use the computer's internal clock. If the language does not have a way to directly access the clock, like BASIC's TI$ or COMAL 2.0's TIME, then use zero-page memory locations 160-162 ($A0-$A2). POKE them with zeros just before the main body of the benchmark and calculate the time at the end with:

```
PRINT (256*256*PEEK(160)+256*PEEK(161)
+PEEK(162))/60
```

8) For more reading on benchmarks, try: "Leaks Like a Sieve," *Byte*, 8/85, p. 33; "Benchmarks," *Byte*, 8/85, pp. 132-3; Gilbreath, J., "A High-Level Language Benchmark," *Byte*, 9/81, pp. 180-198; Knuth, D., *The ARt of Computer Programming Vol 2: Semi-Numerical Algorithms.* Reading MA: Addison-Wesley, 1969; Ahl, D., "Creative Computing Benchmark," *Creative Computing*, 1/84, p. 5.

Finally, let's return to Son of Sieve and I'll get around to answering Robert Boomers' question about COMAL's speed. He sent the following BASIC program that computes the first 1000 prime numbers.

```
10 TI$="000000":DIMY%(8000):P=1:C=0
20 PRINTP,:C=C+1:IFC=1000THEN50
30 P=P+1:IFY%(P)=1THEN30
40 FORX=PTO8000STEPP:Y%(X)=1:NEXT:GOTO20
50 PRINT:PRINTTI/60
```

Translated into COMAL 2.0, Boomers' benchmark reads:

```
10 TIME 0
20 DIM y#(8000)
30 p:=1; c:=0
40 REPEAT
50   PRINT p;
60   c:=c+1
70   REPEAT
80     p:=p+1
90   UNTIL y#(p)<>1
100   FOR x:=p to 8000 STEP p do
110     y#(x)=1
120   ENDFOR x
130 UNTIL c=1000
140 PRINT TIME/60
```

Boomers sent in times for BASIC 7.0 (1 MHz) and Abacus BASIC (MHz unknown), to which I've added a few:

Son of Sieve

	1 MHz	2 MHz
BASIC 7.0	266 seconds	127 seconds
BASIC 2.0	209 seconds	102 seconds
COMAL 0.14*	197 seconds	95 seconds
COMAL 2.0	99 seconds	48 seconds

*COMAL 0.14 is short on program memory (under 10K), so times are estimated based on a benchmark to find the first 725 primes.
**This "run" time does not include the extra step of compiling the program.

So, what do all these numbers mean? You're free to draw your own conclusions. To me they say that COMAL 0.14 is often an improvement over BASIC (in speed, that is; in other ways, it's always an improvement). And COMAL 2.0, depending on your programming application, can have your C-64 running at C-128 "turbo" speeds. You're free to draw your own conclusions. Let me know what they are at P.O. Box 1544, Tallahassee, FL 32302. □

TOMB OF HORROR

The Ultimate 3-D Adventure for the C-64

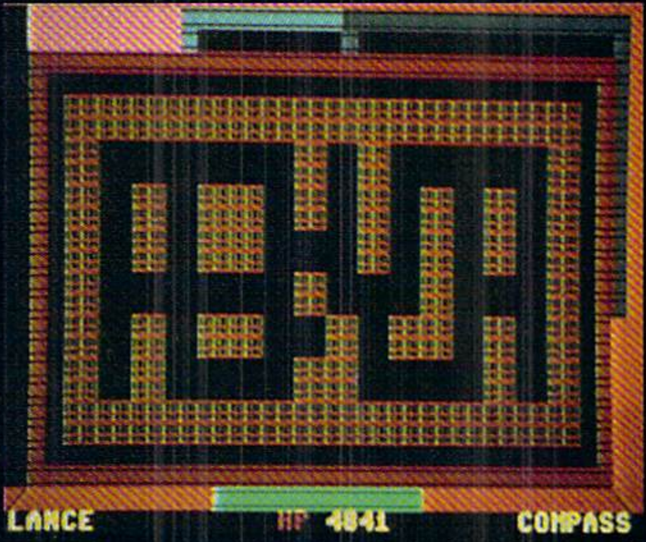
By Cleveland M. Blakemore

At the farthest corner of the earth, beyond the barren wasteland, lies the Tomb, abode of the hideously evil Demon King and the hordes of darkness. The Demon's greatest treasure is the Alchemy Stone, an artifact from the ancient world that transmutes matter itself.

Your task is to retrieve the Alchemy Stone (along with all the treasure you can gather) from the nether regions and escape from the *Tomb of Horror* into the light of day.

This is the third (and final) entry in my trilogy of 3-D games. It evolved out of *Crypt of Fear* (Feb. '88 *Ahoy!*), which in turn was inspired by *Vault of Terror* (Oct. '86 *Ahoy!*). I like to think I've outdone myself, but you will ultimately be the judge of that.

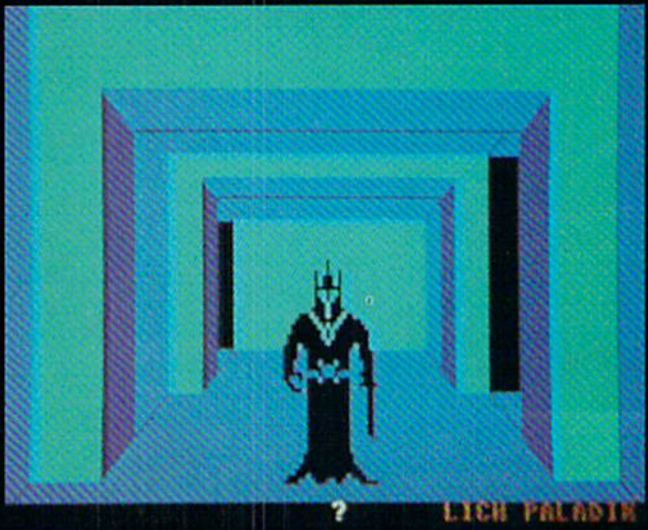




This game is a real-time three-dimensional dungeon adventure that utilizes page-flipping animation, multiple sprite shapes, pull-down windows with joystick-controlled menus, and thirteen different creatures. It has three separate levels, five different kinds of traps, nine different types of weapons, five types of armor, and thirty individual objects that can be held, examined, or used in some way. It also features high quality graphics in color, thanks to extended background mode. The game has complex algorithms for combat, taking into account the type of creature you are fighting, the type of armor you have on, what type and quality weapon you are using, and how strong you are as a fighter.

In order to squeeze this program into a measly 68 blocks on disk, I had to throw all frills to the wind. Although it originally had an attractive title screen and a redefined character font, I had to take these out because of space limitations. In order to assure that the program quality did not suffer one iota, I also had to keep the sound effects to a minimum. I doubt if these deletions will be missed. *(Ed. note: the July* Ahoy! Disk *contains an expanded version of* Tomb of Horror, *complete with redefined graphics and enhanced title screen. See page 52 for ordering information.)*

The program is entirely joystick operated. You will never have to touch the keyboard after typing RUN. All commands are entered through a convenient system of pull-down



THREE WAYS TO SUBSCRIBE TO

COMMODORE USERS **Ahoy!** and **Ahoy!'s AmigaUser**

Any way you stack it up, *Ahoy!* (for Commodore 64 and 128 users) and *Ahoy!'s AmigaUser* offer the best value in news, reviews, features, and programs for the home computerist.

Use the postpaid card bound between pages 50 and 51, or the coupon below, to order 4 or 8 issues of *Ahoy!'s AmigaUser* (published quarterly), 12 or 24 issues of *Ahoy!* (published 8 times a year), or one or two full years of both magazines (4 issues of *Ahoy!'s AmigaUser* and 8 issues of *Ahoy!*)—all at substantial savings over the cover price!

- ☐ **12 issues of *Ahoy!* for $23.00 ($30.00 Canada and elsewhere)**
- ☐ **24 issues of *Ahoy!* for $44.00 ($55.00 Canada and elsewhere)**
- ☐ **4 issues of *Ahoy!'s AmigaUser* for $8.00 ($10.00 Canada and elsewhere)**
- ☐ **8 issues of *Ahoy!'s AmigaUser* for $15.00 ($19.00 Canada and elsewhere)**
- ☐ **8 issues of *Ahoy!* and 4 issues of *Ahoy!'s AmigaUser* for $23.00 ($30.00 Canada and elsewhere)**
- ☐ **16 issues of *Ahoy!* and 8 issues of *Ahoy!'s AmigaUser* for $44.00 ($55.00 Canada and elsewhere)**

Name______________________________

Address____________________________

City______________ **State**______ **Zip**______

Send coupon or facsimile to: ION INTERNATIONAL INC.
45 West 34th Street—Suite 500, New York, NY 10001

menus that erase themselves after use. The commands are simple and straightforward, requiring very little explanation.

To move forward in the dungeon, push the joystick forward. The word "MOVE!" will appear at the bottom of the screen, there will be a slight pause, and you will be standing one block forward from where you were. To turn left, move the joystick left. To turn right, move the joystick right. To move back, pull the joystick back.

If the block you are standing in has anything in it, the bottom line of the screen announces it at the left. If there is a monster in the block with you, the bottom line tells you at the far right.

To access the main menu, you must be standing in a block not occupied by a monster. To pull the menu down, push the fire button and simultaneously pull back on the joystick. The menu appears in blue on the screen, with a selection of thirteen commands to choose from. No creature can attack while you're in the menu.

You can move from one selection to another by moving the highlighted choice in white with the joystick. Push the fire button to select.

The thirteen commands in the main menu are TAKE, DROP, LEFT HAND, RIGHT HAND, ARMOR, ASCEND, DESCEND, EXAMINE, INCANT, QUAFF, SEARCH, INVENTORY, and EXIT MENU. The last erases the main menu window and returns you to real time adventure in the dungeon. Otherwise, you can push the fire button whenever you are finished reading a message or looking at a sub-menu display window, in order to close it and return to the main menu.

TAKE and DROP allow you to pick up and dispose of objects for your inventory. TAKE automatically picks up whatever is in the block with you, providing you do not already have one. DROP allows you to select from one of the four classes of objects you can carry, which read on the sub-menu as WEAPON, ARMOR, SPECIFIC ITEM, and TREASURE. The last entry is EXIT MENU, in case you change your mind and need to escape from the sub-menu.

LEFT HAND allows you to locate a weapon from your inventory and place it in your left hand, your fighting appendage. RIGHT HAND allows you to hold an object from the SPECIFIC ITEMS menu of your possessions. ARMOR permits you to select what you will be wearing. Remember, if you have not selected it from the menu, you are empty handed, no matter what you may be carrying. It is possible to be beaten to death by a flying bat if you have a bazooka in your inventory but have not placed it in your left hand so that you can fight with it.

These commands are very versatile, and many of them are interrelated. You must be holding the compass, for instance, in order to know what direction you are facing.

ASCEND and DESCEND allow you to do just that, if you come across a shaft. You'll need a rope and a steel grappling hook before you can climb up or down.

EXAMINE is also a very versatile command, which you use to look closely at an object in the inventory. You can compare weapons and armor and see which is of better quality. For example, although common sense might dictate that a hand axe would be a better weapon than a dagger, a dagger of excellent workmanship would probably be superior to a hand axe of mediocre or common workmanship. It is left to the discretion of the adventurer. Other objects can also be examined, with various results. I'd like to leave secrets for the player to discover, rather than spoil the fun.

Anything that can be held in the right hand can be INCANTed, which means calling upon its mystical or arcane properties. Don't expect much from INCANTing a compass, but you might try INCANTing suspicious items like the mithril ring (assuming you are in the right locations). INCANT has different effects, depending on the artifact.

QUAFF means to take a swallow of healing elixir, which you carry in one pint servings. It gives you strength for battle and may save you life at an opportune time.

SEARCH looks for traps four blocks forward of your current location. Use this command frequently, or you'll find yourself falling into spiked pits, getting impaled on spears, and becoming tangled in tripwires. Your adventure will be cut short quickly if you are not cautious.

INVENTORY prints out four windows displaying everything you have.

Remember, your long-term goal in the *Tomb of Horror* is to acquire the Alchemy Stone, and all the options on the menu can assist you toward this end one way or another.

Once you have obtained the Alchemy Stone, return to the surface world by ASCENDing through the shaft on the first level in the far northwest corner, and win the game.

I'm certain the game will provide you with hour after hour of entertainment. The quickest game I've played successfully required 45 minutes to get the Stone and return to the upper level.

The whole window system used in *Tomb of Horror* would not be possible without the use of Shawn K. Smith's clever memory move command from the June '87 *Tips Ahoy!*, which uses the built-in ROM routines to stash and save text and color information in a buffer above BASIC memory set aside for the purpose. My thanks to Shawn for the use of his short ML tip.

The game is in two parts: a BASIC section and a binary file. Be certain that the *Flankspeed* file is named correctly so that the main program can load it into memory.

Tomb of Horror contains numerous shifted spaces which must be typed in correctly, because the program uses extended background mode and requires shifted characters to generate different colors. Be especially careful when typing in the game.

Giving credit where credit is due, I'd like to pay tribute to the game *Dungeons of Daggorath* for the Tandy TRS-80 as the driving force that compelled me to create my own 3-D games for the C-64. Ever since I saw this game back in the summer of 1985, I've been striving to create a 3-D game as good or better.

If you think I succeeded, or failed miserably, please let me know. I sincerely enjoy reading mail, both critical and flattering, and would like to hear what all *Ahoy!* readers think of *Tomb of Horror*, or any other games I've created in the past. Hearing the voice of the readers gives direction to my efforts. If you would really like to see "just one more" 3-D game, or hope I'll never waste so much magazine space again, let me know, c/o *Ahoy!* You might see one of your suggestions incorporated into a game in the future! ☐ ***SEE PROGRAM LISTING ON PAGE 72***

Continued from page 23

an aircraft carrier! Better still, *Jet* interfaces with the entire library of *FS* Scenery Disks (as does this new program), so computer pilots can wage contemporary high-speed air combat over just about any city on Earth.

Stealth Mission, written by Steve Setzler and "directed" by Bruce Artwick, represents the latest wrinkle in the ongoing saga of the software Icarus. Unlike *Jet*, it does not contribute a quantum leap forward in terms of design. Technologically, however, it pushes the Commodore 64/128 envelope well beyond the blue horizon, to a whole new level of animation and frame rates.

The visual integrity in all cockpit displays is absolutely incredible. The program never freezes in order to load new data, but rolls smoothly with the exception of occasional color flip-flops.

The emphasis here is not so much on the simulation as on the flight and combat elements. The fact is, the aircraft being simulated here – F-19 Stealth Fighter, Navy F-14 Tomcat, and the experimental forward-swept wing X-29 – could probably be flown by chimpanzees. These aircraft do everything but reproduce themselves. SubLOGIC made the design almost idiotproof. Single button commands allow the fighters to locate and rendezvous with a refueling plane. There's even an autolanding feature. All this flight lacks is Coca-Cola in the cockpit.

Stealth Mission offers eight scenarios, selected from an onscreen menu. These missions range from easy to intermediate to advanced, and include a mountain conflict, a battle at sea, a marathon bomber scenario, and a visually appealing mission in which the air combat occurs over a collection of ancient ruins.

A second screen allows the user to set difficulty (0-9), select an aircraft, and choose its armament (Sidewinder, Sparrow, and Maverick missiles, four kinds of bombs, AGM-84 Harpoon, air-to-air and air-to-ground Stealth missiles, and an M61 cannon).

Instrumentation includes all the standard items (HUD, air speed indicator, altimeter, fuel level, thrust, clock, brake and landing gear status, altitude indicator, etc.). There are also navigational aids (ADF receiver and bearing indicator, VOR receiver and omnibearing indicator, DME and ILS glideslope needle) and weapons/defense controls (damage indicator, radar, target-tracking computer, ECM, weapons select, etc.).

The controls will seem immediately familiar to anyone who's ever used a flight simulator (or, heck, even a real jet): throttle, elevator, aileron controls, airbrakes, landing gear, etc. *Stealth Mission* also allows for joystick control of flight operations.

In addition to the standard *FS/Jet* options (multiple viewpoints, zoom, control tower view, spotter plane), this program adds a "missile's eye view" to intimately track the killer projectiles as they zoom toward a computer-locked target.

Is *Stealth Mission* exciting? Yes, but not as exciting as it should be. The problems are almost intrinsic to the crafts being simulated. They are easy to fly and the onboard computers stabilize them quickly when danger approaches. As a result of all this convenience, the player doesn't have much to do for long stretches of time. The upside, of course, is that users are free to concentrate on combat. However, these air battles have a cold, somewhat antiseptic feel. Instead of simply lining up a target in your sights and blasting away, the computer must be locked on before a weapon is discharged, and that won't happen until the target is within range.

The weaponry is impressive as all getout, with a choice of 11 different missiles, bombs, and cannon, but the results are less than satisfying.

With all the visual marvels SubLOGIC produced for this game, it's a shame more time wasn't spent simulating the *results* of the warfare. Explosions are nothing special, from either a visual or audio perspective, and we aren't even treated to the sight of an enemy aircraft crashing in flames!

The overall package, however, is excellent. The game comes with maps, extensive and well-written documentation, and a quick-reference card. *Stealth Mission* is an incredible programming accomplishment that deserves (and will doubtless have) the attention of anyone interested in the state of the combat flight simulation art.

SubLOGIC Corp., 713 Edgebrook Dr., Champaign, IL 61820 (phone: 217-359-8482). *–Bill Kunkel*

Ahoy! DISK

Why type in the listings in this month's *Ahoy!* when we've done it for you? All the programs in this issue are available on the current *Ahoy! Disk* for $8.95. Isn't your time worth more than that?

Use the coupon at the bottom of this page to order disks for individual months, a disk subscription, or the special anthology disks described below. (You can also subscribe to the *Ahoy! Disk Magazine*—disk and magazine packaged together at a special reduced rate. See the card bound between pages 50 and 51.)

BEST OF '84
Address Book (3/84)
Air Assault (5/84)
Alpiner (6/84)
Checklist (7/84)
Construction Co. (12/84)
DOS (9/84)
Emerald Elephant (10/84)
I Got Biorhythm (5/84)
Lawn Job (10/84)
Lunar Lander (4/84)
Math Master (5/84)
Micro Minder (8/84)
Multi Draw 64 (1/84)
Name That Star (4/84)
Post Time (6/84)
Salvage Diver (9/84)
Space Lanes (VIC) (3/84)
Space Lanes (64) (3/84)
Space Patrol (12/84)
VIC 40 (10/84)

BEST OF '85
Ahoy! Dock (9/85)
Auto Line Numbers (3/85)
Directory Manipulator (12/85)
File Lock (9/85)
Fontasia (2/85)
Futurewar! (2/85)
Gameloader (12/85)
Home Budget (3/85)
Jewel Quest (12/85)
Lightning Loader (11/85)
Midprint (1/85)
Music Tutor (1/85)
Salmon Run (3/85)
Screen Dump (8/85)
Telelink (2/85)
Tile Time (5/85)

BEST OF '86
Disk Cataloger (8/86)
Dogcatcher (9/86)
Fidgits' Music School (7/86)
Flap! (8/86)
Guardian (11/86)
Head to Head (2/86)
Highlight (8/86)
JCALC (4/86)
Meteor Run (8/86)
Mine Canyon (9/86)
Mountaineer Mack (10/86)
Star Strike (6/86)
Step On It (8/86)
Syntax Patrol (9/86)
Teleporter (11/86)
Tenpins (11/86)
1541 Alignment Check (2/86)

BEST UTILITIES
Auto Append (7/85)
BASIC Trace (4/85)
Base Conversions (9/84)
Booter (4/85)
Commodore Font Editor (5/85)
DOS (9/84)
Fastnew (9/85)
Formatter (8/85)
Lucky Lottery (7/85)
Maestro (10/85)
Micro Minder (8/84)
Printat (7/85)
Screen Bit Dump (6/85)
Screen Magic (6/85)
Start & End File Address (9/85)
Two Column Directory (5/85)
Windows (8/85)
1541 Cleaning Utility (10/85)

BEST UTILITIES II
Alarm Clock (1/86)
BASIC Relocator (1/86)
Bigprint (5/86)
Chrono-Wedge (4/86)
Disk Checkup (10/86)
Failsafe (5/86)
Long Lines (11/86)
Programmer's Aid (3/86)
Quick Change (10/86)
Reversed Remarks (7/86)
Screen Sleuth (7/86)
Screen Window (4/86)
Speech64 (9/86)
The Editor (12/86)
Trackdown (6/86)
Very-Able (3/86)
128 to 64 Autoboot (10/86)

BEST GAMES
Alpiner (6/84)
Construction Co. (12/84)
Disintegrator (8/85)
Dragon Type (9/85)
Duck Shoot (6/85)
Emerald Elephant (10/84)
Gators N Snakes (8/85)
Gravinauts (10/85)
Lawn Job (10/84)
Mystery at Mycroft Mews (10/85)
Name That Star (4/84)
Post Time (6/84)
Salvage Diver (9/84)
Script Analysis (7/85)
Space Patrol (12/84)
Tile Time (5/85)
Wizard of Im (7/85)

BEST GAMES II
Actionauts (4/86)
Air Rescue (4/86)
Brickbusters (3/86)
Crabfight (2/86)
English Darts (5/86)
Got a Match? (6/86)
Haunted Castle (1/86)
Head On (7/86)
Knockout (1/86)
Micro City (12/86)
Minotaur Maze (12/86)
Penguins (10/86)
Ski Folly (5/86)
Star Search (5/86)
Swoop (4/86)
The Last Ninja (9/86)

BEST C-128
Ahoy!Term (3/86)
Dark Fortress (1/87)
Dual Dump (9/86)
Extended Background Mode (4/87)
Infoflow (4/87)
List Formatter (4/87)
More BASIC 128 (2/87)
Nebergall Run (7/86)
Planet Duel (4/87)
Queen's and Bishop's Tours (6/86)
Rescue 128 (3/87)
The Artist (12/86)
Variable Manager (9/86)
Wordcount (7/86)
128 to 64 Autoboot (10/86)

We've collected Ahoy! 's best programs onto the Anthology Disks described here. Please note that many of these programs will be unusable without the documentation printed in the issues of Ahoy! listed in parentheses.

SINGLE ISSUE AND SUBSCRIPTION PRICES

(Postage and Handling Included)

	U.S.A.	CANADA	ELSEWHERE
❑ July '88 *Ahoy!* Disk	$ 8.95	$ 12.95	$ 13.95
❑ 12-Month Disk Subscription	$ 79.95	$ 99.95	$124.95
❑ 24-Month Disk Subscription	$149.95	$179.95	–

BACK ISSUES $8.95

In Canada add $2.00 per issue; outside US and Canada add $4.00 per issue.

❑ ____	$8.95	❑ ____	$8.95
❑ ____	$8.95	❑ ____	$8.95
❑ ____	$8.95	❑ ____	$8.95
❑ ____	$8.95	❑ ____	$8.95

ANTHOLOGY DISKS $10.95 EACH

❑ Best of '84 ❑ Best Utilities
❑ Best of '85 ❑ Best Utilities II
❑ Best of '86 ❑ Best Games
❑ Best of C-128 ❑ Best Games II

Send coupon or facsimile to: *Ahoy!* Program Disk
Ion International Inc.
45 West 34th Street, Suite 500
New York, NY 10001

MC and VISA Accepted Call 718-879-9330

NAME ____________________
ADDRESS ____________________
CITY ____________ STATE ______ ZIP ______

Make check or money order payable to **Double L Software**. N.Y. State Residents must add 8¼% sales tax.

geoCALC 128
Berkeley Softworks
Commodore 128
Disk; $69.95

Berkeley has been busy evangelizing the Commodore world on the benefits of their Graphic Environment Operating System *(GEOS)*. Part 2 of that mission has been to provide a bevy of applications which utilize the power of the icon. With that in mind they have released a series of C-128 products which take advantage of the machine's inherent advantages over the 64.

One of those releases is *geoCalc 128*, the graphic interface spreadsheet which accepts worksheets produced by its C-64 counterpart. This copy protected disk requires *GEOS 128* version 1.2 or later, a 128, an 80 column RGBI monitor, a joystick or a mouse, and a disk drive (1541 or 1571).

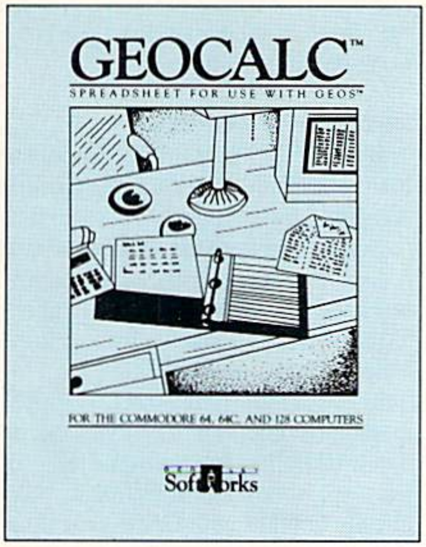
Optional equipment includes a printer, a second disk drive, and a 1750 RAM Expansion Unit (REU). Everyone understands the benefits of a printer and a second disk drive–hardcopy and increased file efficiency–but the REU is another matter. With it, spreadsheets recalculate faster, and larger spreadsheets are possible; therefore less time is spent waiting. That's the whole reason for using computers in the first place.

The Basics

Though copy protected, the application provides a way to make backup copies, the actual work disks. As with most things *GEOS* it is usually best to stay with the routines provided on the disks, whether you're duplicating entire disks or merely copying files.

gC allows a worksheet of up to 256 rows by 112 columns for a cell capacity over 28,000. It provides 12 digit accuracy for addition, subtraction, division, multiplication, and percentage calculations, and 9 place accuracy for trigonometric, exponential, logarithmic, and financial (present/future values, etc.) operations. Up to 200 characters can reside in a single cell.

Almost half of the menu options have keyboard alternatives; the screen display is logical and well-executed.



Useful for light duty jobs at best.
READER SERVICE NO. 177

Cell widths can be changed from 3 to 31 characters, with 12 being the default. Relative and absolute references are supported, as are named ranges and the ability to copy, cut, clear, or paste data (and functions) within a sheet.

Just over a dozen display formats are available ($0.0, 0.00, 0.0000, for example); and information can be imported or exported to other *GEOS* applications via the scrap function. Alphanumeric alignment functions include centering in a cell, and left or right justification. Character attributes can be displayed and printed as italic, bold, bold-italic, or plain.

Advanced Features–Somewhat

geoCalc 128 handles functions such as ATAN (arctangent), COS (cosine), LOG (logarithm), RAND (random), SIN (sine), SQRT (square root), and TAN (tangent), among others. These, and the other 16 functions, represent the best the program has to offer, mathematically speaking.

gC does make use of the 128's uniqueness–somewhat. The numeric keypad is supported; likewise, its ENTER key functions the same as the RETURN key. The individual cursor keys work, but the function keys are sorely neglected. So is the NO SCROLL key and the HELP key. (There is no online help, and judging from the condition of my manual, there should be.)

The ability to access an REU's extra memory is excellent, however, as are the numerous ways (five!) to move the cursor about the worksheet.

Entire rows or columns can be selected by clicking on the row or column's alphabetic or numeric label. That is, all the cells in column B can be highlighted (ranged) by moving the cursor over the "B" and clicking the left mouse button. The same applies to numeric rows.

Manual Amnesia

The Dr. Jekyll/Mr. Hyde mix continues as the manual covers both the 64 and the 128, with 128-specific info noted in highlighted boxes. But all the screen illustrations, and there are plenty, are from the 128 version.

Chapter 3 is a tutorial of sorts. Chapter 4 continues the process in greater depth until, in my manual anyway, page 4-6 is reached. That's where Chapter 4 ends abruptly. Full explanations of the "advanced" functions and other bits of information are supposed to be on pages 4-7 to 4-41, give or take a page or two. (Supposed existence of the other pages was revealed by scanning the index for the whereabouts of information concerning the trigonometric functions.)

An outright error is displayed in the screen graphic on page 4 in Chapter 3. A call-out arrow points to the checkmark entry icon, calling it the text entry cursor. The cursor is nowhere to be found, further confusing the issue and making it more difficult to determine the true state of affairs.

Other Anomalies

Some functions are only available via the keyboard, a strange turn of events considering the entire reason this program exists–a graphic interface with icons and pulldown menus.

One such option is the cancel command, useful to abort an incorrect procedure or calculation; the other facilitates marking large chunks of cells for range manipulations. Both are important by any standards.

The fact that there are two data win-

dows available is nothing new either. Many spreadsheets have the ability to split a display into an active an an inactive window, and then scroll through the active one. *gC* only allows horizontal splits–that's like building half a house.

To install *geoCalc 128* with a 1571 drive, the software resets it to act as a 1541. While the procedure is noted in the manual, the need for this kind of manipulation lends credence to the belief that *gC 128* is nothing more than a warmed over *gC* for the 64. That would be okay if all of the 128's unique features were addressed, but they're not.

The program is described as being "intelligent," a phrase usually reserved for spreadsheets which use a sparse memory matrix to recalculate only cells containing entries. This method produces faster results; however, it is not supported by *geoCalc 128.* Nowhere is the term "intelligent" properly described or explained.

After making a big to-do about the high quality of the printed output, it seems strange that the Apple LaserWriter is not supported. After all, several other *GEOS* products include drivers for this standard setting printer. Lack of a device driver for this high-end printer muddies the water over the true level of the program's "presentation" quality printing.

How Does It Add Up?

geoCalc 128 is a fine concept, but it has very little mathematical muscle. It cannot import other spreadsheet files (beyond those from the C-64 *geoCalc*); it does not support truly advanced features and functions (macros, conditional searches, etc.); and the manual is erroneous and incomplete.

Its prime attraction is *GEOS's* graphic, intuitive interface. This "window on the world" provides extreme ease of learning, a comfortable feeling if you're already familiar with *GEOS* programs, and ease of use.

Unfortunately, in this case, that is not enough. Mice, pulldown menus, and icons do not make a program; they are merely tools which drive or feed an application. The software must do something substantial, must provide a cost-effective solution to a need. In those regards, *geoCalc 128* is not much more than a pretty face. At best, it is useful for light duty jobs.

Berkeley Softworks, 2150 Shattuck Avenue, Berkeley, CA 94704 (phone: 415-644-0883). *–Ted Salamone*

TISAR 128
CW Data Labs
Commodore 128
Disk; $89.95

The first thing you should understand about *TISAR 128* is that TISAR is an acronym that stands for Total Information Storage And Retrieval. Having that as its name and its apparent goal, you'll probably guess that it is a database. More than that, however, it is a database that has been optimized for small business use–one that will create and print point-of-sale invoices, database records with monthly or yearly totals of accounts received and receivable, and a perpetual inventory based on sales.

As such, it is one of the most powerful programs I've seen for the C-128, even though it will not have an audience as large as would be the case with a "free-form" relational database.

And for those who may despair because they've been told graphics are all but impossible in the C-128's 80-column mode, we'll point out that *TISAR 128* uses them almost to a fault: simple graphics, true, but eye-catching, as is their intent.

Although it is possible to pick one or another section of *TISAR 128* and use it exclusively, the heart of the system as designed is the database. Since we must use the Commodore 1571 disk drive with double-sided disks, it is possible to install the database program and up to 500 records on a single disk. (Almost necessary for efficiency, since *TISAR* supports only one drive.)

Depending upon whether your business is sales/service or sales/marketing, you'll pick the appropriate program for the job. *TISAR 128* supplies you with two service disks and one marketing disk. With no ability to edit or customize database fields, it is important that variations are provided, but you may still have to bend your current record-keeping methods to fit.

Fields provided are for name, address, phone, part number, remarks,

estimated price, deposit, and COD. If, for example, an item is purchased for $500.00 and a deposit of $100.00 is made, the larger amount is entered as Estimate. COD entries occur only when the balance is received. In this way, *TISAR* allows you to see at a glance the amount of money received for a particular period (month or year) as well as money owed you for that same period.

It does this through one of the many search features—records can be searched on any field and printed or displayed. Additionally, there is a cross-search feature which allows sorts on multiple fields. As an example, if you wanted to know how many people named Smith had bought widgets during the month of March, you could.

In screen displays, *TISAR* will present the records in a scrolling form with outstanding amounts highlighted in red, or as a bar chart in which the amounts received will help you to get a handle on the month-to-month ups and downs of your business.

Unfortunately for its stated use as a point-of-sale invoicing system, a database record must be created before an invoice can be filled out. While this would be fine if all customers were repeat customers—and thus a database record on each already existed—I feel that here the long process would slow things to the point that you'd be losing customers. As an alternative, there might be some value in using *TISAR* as a "back-room entry" system, where the true accounting is done after the sale and the invoice would be mailed to the customer simply as a confirmation of the transaction.

In beginning an invoice, you would first probably choose to customize it by having the printed copy show your business name, address, and phone number. As is the case with each of its many routines, *TISAR* will prompt you for each separate entry, and many times will do this by creating a flashing window on the screen.

Following this is a section for printing your company's guarantee or any other message you may need to impart: "No refunds after ten days," "Repairs left over thirty days," etc.

The invoice form itself will depend upon whether your business is sales and marketing or sales and service. As it does with the databases, *TISAR 128* provides a different invoice for each type of business. When dealing with invoices—as opposed to database records—it is necessary to have several formatted disks on hand. Each disk will hold 125 invoice records, and this in spite of the fact that they are double-sided disks formatted in the 1571 drive.

At the point of sale, enter the customer name; quantity, part number; description of goods or services; price per unit; sales tax rate (or tax exempt); and the amount of deposit or payment. All this is in answer to screen prompts, and fields that do not apply can be avoided by entering "0". Once all entries have been made, the invoice, complete with subtotal, total, deposit, and balance will be printed to the screen and can be sent to your printer, where the heading and guarantee messages will also be printed.

The third part of the system is called TISAR ICM—Inventory Control Module. With this, you can store records of up to 1000 individual items on a disk, and there are fields for adding to stock as new merchandise is received or deleted from stock as items are sold.

To accomplish the latter, ICM will process the invoice data disks, reading the items sold and subtracting them from inventory, allowing you a much better feel for the pulse of your business—allowing you to know what items are selling, when they are selling, and the rate of movement. In addition, each item can be keyed to a reorder point: when only 50 widgets are left in inventory, it is time to order more.

With selective display and printing options, the ICM module will allow you a look at all inventory items or only those that have reached the reorder point.

There is no doubt that *TISAR 128* can be of value to small businesses in helping them to gain better control of the record-keeping and inventory processes, but it does have shortcomings.

First, with the number of disks and programs involved, it should have been written for use on a hard disk or, at the very least, to support two disk drives. Second, it is a bit too cumbersome to be used as a point-of-sale system, though this could have been alleviated by turning the program around. If the database were created from the invoice, it would be a much faster sys-

Continued on page 81

COMMODARES
PROGRAMMING CHALLENGES

By Dale Rupert

Each month, we'll present several challenges designed to stimulate your synapses and toggle the bits in your cerebral random access memory. We invite you to send your solutions to:

Commodares, c/o *Ahoy!*
P.O. Box 723
Bethel, CT 06801

We will print and discuss the cleverest, simplest, shortest, most interesting and/or most unusual solutions. Be sure to identify the *name* and *number* of the problems you are solving. Put *your* name and address on the listings as well. Show sample runs if possible. Briefly describe your solutions and tell what makes them unique or interesting, if they are. You must enclose a stamped, self-addressed envelope if you want any of your materials returned. Solutions received by the middle of the month shown on the magazine cover are most likely to be discussed, but you may send solutions and comments any time. Your original programming problems, suggestions, and ideas are equally welcome. The best ones will become *Commodares*!

PROBLEM #55-1: SIG DIGS

This problem was submitted by Jim Speers (Niles, MI). Write the shortest program which accepts any integer from 0 to 32767 and prints out the number of 1's in its binary representation. The title of this problem comes from the more advanced challenge which is to accept any whole number from 0 to 65535 and print out the number of 1's, significant 0's (ignore leading 0's), and total number of SIGnificant DIGits in its binary representation. (Jim thinks that this second challenge cannot be done in one line.)

For example, if the user enters 65, the computer responds "2" for the first challenge, since the binary value of 65 is 0100 0001 which contains two 1's. For the second challenge, the computer responds "2, 5, 7" since there are two 1's, five non-leading 0's, and seven significant digits.

PROBLEM #55-2: RAPID RANDOMIZER

Try this one from Justin Smalley (Boulder, CO). Start with the following array definition:

```
10 N=100 : DIM X(N): FOR I=1 TO N: X(I)=
I: NEXT
```

Write the rest of the program which will rearrange this array in random order as quickly as possible. Add line 20 to start the timer and line 40 to stop the timer and print the time required to shuffle the array.

```
20 T0 = TI
40 PRINT TI-T0 "JIFFIES"
```

Start your solution at line 30.

PROBLEM #55-3: UNJUMBLED WORDS

Eddie Byrd (Ste. Genevieve, MO) wants to see every possible combination (or permutation) of the letters of any five- or six-letter word entered by the user. He wants his computer to help solve those scrambled-word puzzles in the newspaper. Can you help? (Readers with a good collection of *Ahoy!* back issues may not have much trouble with this one.)

PROBLEM #55-4: POLAR GRAPHICS

This one from Thomson Fung (San Diego, CA) should not be too difficult if you have a C-128 or use *Simon's BASIC* on the C-64. Very simply, write a program to plot a graph of the equation

```
R = 10*T*COS(T)*SIN(T)
```

in polar coordinates. Here R stands for the distance of each point from the origin. T represents the angle of a line from the origin to the given point. If you let T go from 0 to 25.2 in steps of 0.1, you will be rewarded with an interesting butterfly.

First of all I must confess to a blunder regarding the discussion of *Problem #48-4: More Income* in the April issue of *Ahoy!* The problem was the math classic where you are given the option of receiving $15000 per year under one of two plans. Plan A gives a raise of $750 every six months. Plan B gives a raise of $3000 every year. The challenge was to write a program to display the annual and cumulative pay for the two plans. I chastised the readers for their confusion on this problem, since it seemed that most people misinterpreted it.

Fred Simon (Lindenwold, NJ) sent me a letter stating that my readers were not the only ones confused. Although I disagree with Fred's interpretation of the problem, it caused me to rethink the whole situation. Here (I think) is the proper answer to the problem. Plan A is better if you work for just one year; otherwise Plan B is better. (I originally said that Plan A was significantly better overall, contrary to common sense.)

Analyze it this way. Every six months, Plan A increases your base annual salary (BAS) by $750. Your pay for any six month period is the current BAS divided by two. For example, your BAS for the first six months is clear-

ly $15000, and your six months' salary is $15000/2 or $7500. Everyone agrees on that.

At the end of six months, your BAS goes up to $15750, so your pay for the second six months is $15750/2 or $7875. Your total first-year earnings are $15375 ($7500 + $7875). After one year, your BAS is raised another $750 to $16500. You earn $16500/2 ($8250) for the first six months of the second year. Then your BAS is raised to $17250, and you get $17250/2 ($8625) for the fourth six-month period. Your total over two years is $32250.

With Plan B, your BAS is $15000 for the first year, and that is what you receive. Your BAS is raised to $18000 which is your second year's salary. At the end of two years, you have accumulated $33000. Plan B income is ahead of Plan A income by the middle of the second year.

Here is Fred's program adapted slightly to agree with this analysis.

```
•1 REM ====================================
•2 REM     COMMODARES PROBLEM #48-1 :
•3 REM            MORE INCOME
•4 REM ====================================
•10 PRINT"YEAR SALARY A TOTAL[3" "]SALARY
   B TOTAL"
•20 BAS=15000: SA=BAS: SB=BAS: TA=0: TB=0
•30 RA=750: RB=3000
•40 FOR YEAR=1 TO 10
•50 TA=TA+SA/2+(SA+RA)/2
•60 TB=TB+SB
•70 PRINT YEAR;TAB(5);"$"SA;"$";TA;
•80 PRINT TAB(22);"$"SB;"$"TB
•90 SA=SA+2*RA: SB=SB+RB: NEXT
```

SA and SB are the BAS for Plan A and Plan B. TA and TB are the cumulative totals.

An interesting aspect of this problem is that Plan A is a better plan if it offers semiannual raises of $1500 as opposed to Plan B's annual $3000 raise. RA in the program is the amount of Plan A's raise. In fact, if you work for less than eight years, you would be better off with Plan A if it offered increases of only $1400 semiannually. (Can anyone figure out the break even point, that is, the amount of Plan A's increase which is equivalent to Plan B?) Experiment with the values and see what you come up with. Thanks to Fred for helping to straighten this out. My apologies to anyone else whose solution was correct the first time around.

Now on to new business. We will look at the best solutions to *Commodares* in the February issue of *Ahoy! Problem #50-1: Phone Decode* was submitted by Pat McConville (Manassas Park, VA). The user inputs a sequence of two-digit numbers representing letters encoded according to the telephone keypad. The computer decodes the numbers into letters. The first digit is the keypad number. The second digit is the letter's position on that key. Since "D" is the first letter on the number 3 key, its code value is 31. "Q" and "Z" are not on the phone keypad, so they are assigned 77 and 99 respectively.

Here is an interesting solution from Eddie Byrd (Ste. Genevieve, MO).

```
•1 REM ====================================
•2 REM     COMMODARES PROBLEM #50-1 :
•3 REM        PHONE DECODE
•4 REM     SOLUTION BY
•5 REM        EDDIE BYRD
•6 REM ====================================
•10 A$="[20" "]ABC[7" "]DEF[7" "]GHI[7" "
   ]JKL"
•20 A$=A$+"[7" "]MNO[7" "]PRS[3" "]Q[3" "
   ]TUV[7" "]WXY[5" "]Z"
•30 INPUT A(1),A(2),A(3)
•40 FOR X=1 TO 3
•50 PRINT MID$(A$,A(X),1);
•60 NEXT:PRINT:GOTO 30
```

Be careful typing the spaces in the definition of A$ since each character's position in A$ matches its encoded value. "D" is the 31st character in A$, and "Z" is the 99th character. The MID$ statement selects the chosen characters, one at a time. Eddie's method handles invalid numbers greater than zero by returning the space character. To avoid an Illegal Quantity error if the user just presses RETURN without entering a number, add this line:

```
45 IF A(X)=0 THEN 60
```

Don Wilkins (Burke, VA) transformed the numerical input into letters with the following formulas:

```
L = 3*INT((N-20)/10)
R = N-10*INT(N/10)
C = L+R-((L+R)>16)+6*(R=7)+5*(R=9)
```

N is the input number (20 < N < 100). The corresponding letter is printed with the statement

```
PRINT CHR$.(64 + C)
```

Don's program first checked for valid inputs. The last two addends in the formula for C correct for the letters "Q" and "Z". Recall that if (R=7) is true, its value is −1. Consequently 6*(R=7) subtracts six whenever R is seven. The middle term in the formula for C corrects all letters above "Q".

Ernest Barkman (Athens, NY) suggested *Problem #50-2: Short Sound.* The idea is to write the shortest program which will generate a continuous sound. Ernest's solution for the C-64 or the C-128 is shown in line 21 of the following program:

```
•1 REM ====================================
•2 REM     COMMODARES PROBLEM #50-2 :
•3 REM        SHORT SOUND
•4 REM     SOLUTIONS BY
•5 REM        ERNEST BARKMAN     (LINE 21)
```

```
·6 REM      ERNEST BARKMAN SR. (22 & 23)*
·7 REM      BOB RENAUD         (24)*
·8 REM      KEVIN ESHBACH      (25)
·9 REM      WALLACE LEEKER     (26)*
·10 REM =====  (* = C-128 ONLY)  ========
 =
·21 POKE 54296,RND(1)*9:RUN
·22 PRINT CHR$(7): GOTO 22
·23 A=RND(1)*7+65:PLAY CHR$(A):GOTO 23
·24 PRINT"[CNTRL G]":RUN 24
·25 FORI=54272TO54296:POKEI,RND(1)*255:NE
 XT:RUN 25
·26 SOUND 1,999,1:RUN 26
```

Calling the output of this program a "sound" is about the most we can do. "Raspiness" might be a more appropriate term. Still, the program takes only 18 keystrokes (using abbreviations, ignoring spaces, and disregarding the line number and the RETURN at the end of the line), and Ernest's challenge was for brevity. Note that this program takes one more keystroke on the C-128 than on the C-64. Why? "P" SHIFT-O" is POKE on the C-64, but it is POT on the C-128. Use "PO SHIFT-K" on the C-128.

This program randomly sets the volume of the sound generator. Why that should make a sound is not clear. Evidently this challenge generated some family rivalry. The solutions in lines 22 and 23 are from Ernest Barkman Sr. (Orlando, FL), either Ernest's father or else quite a coincidence.

Line 24 by Bob Renaud (Washington, MA) is an even shorter version of line 22. The CONTROL-G which beeps on the C-128 is entered in quote mode. Ignoring line numbers at the beginning and the end of the line, this program takes eight keystrokes using "?" for PRINT. This is undoubtedly the shortest program possible for solving the problem.

Not as short but producing more interesting sounds is line 25 from Kevin Eshbach (Sanatoga, PA). It POKEs random numbers into various SID registers. It runs on the C-64 or the C-128. Wallace Leeker (Lemay, MO) sent the program in line 26. The sound is a pure tone with a slight discontinuity each time the program is rerun. Jim Speers sent a similar solution.

To run any of these, just type RUN followed by the appropriate line number. RUN 25 executes Kevin's program, for example.

Now for *Problem #50-3: Tricky Index* from Justin Smalley (Boulder, CO). The problem is to write line 30 to replace line 29 in the following program so that line 40 prints the index vertically in columns rather than in rows as line 29 does. The result is to have the same number of rows and columns as produced by line 29. Justin mentioned that such a routine can be handy for printing out an alphabetized array.

Jim Borden (Carlisle, PA) sent the following solution which takes advantage of the predefined variables. This allows his program to be easily modified for any number of columns NC and any value of L.

```
·1 REM ===================================
·2 REM    COMMODARES PROBLEM #50-3 :
·3 REM       TRICKY INDEX
·4 REM    SOLUTION BY
·5 REM       JIM BORDEN
·6 REM ===================================
·10 L=17: NC=5: NR=INT(L/NC)
·20 FOR R=0 TO NR: K=R*NC: FOR C=1 TO NC
·29 REM I=K+C: IF I>L THEN END :REM GIVEN
·30 E=1-(NC*NR-L):D=C-E:I=R+1+(C-1)*(NR+1
 ):IF D>-1 THEN I=I-D:IF R=NR AND C=E THE
 N END
·40 PRINT I;: NEXT: PRINT: NEXT
```

Jim mentioned that E in line 30 should have been defined in line 10, but the *Commodore* allowed changes only to line 30. Do you see the disadvantage of defining E in line 30? In this program, E will be defined 18 times, even though its value never changes. You should not define a variable inside a FOR-NEXT loop unless its value changes within the loop.

Jim Speers suggested modifying line 40 so the results are printed in uniformly spaced columns. He used this statement on the C-128:

```
40 PRINT USING"###";I;: NEXT:PRINT:NEXT
```

Bob Renaud converted the values to be printed into strings "padded" with spaces to give uniform columns.

Many readers had fun solving *Problem #50-4: Simple Simon.* The challenge was to write the simplest program possible to create the Simon number-sequence memory game. The computer briefly displays three numbers. The user must remember them and enter them in order. The computer then adds another number to the sequence and displays the four-number sequence briefly. The user must enter them in correct order. This process continues until the user fails to enter the list correctly.

This brief program from Judy Groth (Brooklyn, NY) solves the problem in only three lines.

```
•1 REM ==================================
•2 REM     COMMODARES PROBLEM #50-4
•3 REM      SIMPLE SIMON
•4 REM     SOLUTION BY
•5 REM      JUDY GROTH
•6 REM ==================================
•10 DEFFNR(S)=INT((RND(S)*9)+1):FORX=1TO3
 :A$=A$+MID$(STR$(FNR(1)),2,1):NEXT:PRINT
 A$
•20 FORD=1TOX*300:NEXT:PRINT"[CLEAR]"::IN
 PUTB$:IFB$<>A$THENPRINT"SCORE="((X-4)*10
 ):END
•30 X=X+1:A$=A$+MID$(STR$(FNR(1)),2,1):PR
 INT"  "A$::GOTO20
```

Notice Judy's user-defined function in line 10. This function generates random integers from 1 to 9. This program keeps score, awarding 10 points for each digit in the last number entered correctly.

As if three lines were not short enough, Sharon Albers (Walters, MN) sent this nifty two-liner.

```
•1 REM ==================================
•2 REM     COMMODARES PROBLEM #50-4
•3 REM      SIMPLE SIMON
•4 REM     SOLUTION BY
•5 REM      SHARON ALBERS
•6 REM ==================================
•10 X$=X$+RIGHT$(STR$(INT(RND(0)*10)),1):
 ON-(LEN(X$)<3)GOTO4:PRINT"[CLEAR][DOWN][
 DOWN][RIGHT][RIGHT]"X$
•11 FORT=1TO500:NEXT:INPUT"[CLEAR][DOWN][
 DOWN]";I$:ON-(X$=I$)GOTO4:PRINT"INCORREC
 T -- "X$:END
```

X$ holds the computer's number sequence. The first statement in line 10 picks a random digit, converts it into a string, and strips away the leading space. Another interesting aspect of Sharon's program is her use of the ON-GOTO statements. These allow the equivalent of IF-THEN conditional statements to be executed while having other commands follow on the same line. This is a handy procedure for the C-64 which does not allow the ELSE statement. A nice feature of this program is the final PRINT statement which shows the actual number which the user missed.

Some readers were careful to empty the keyboard buffer before accepting user input. This is to prevent cheaters from typing the numbers while they are displayed on the screen. (Of course, the real criminals would use pencil and paper.)
POKE 208,0 clears the keyboard buffer on the C-128. Use POKE 198,0 for the C-64.

The ultimate in brevity is this one-liner for the C-128 from Charles Kluepfel (Bloomfield, NJ).

```
•1 REM ==================================
•2 REM     COMMODARES PROBLEM #50-4
•3 REM      SIMPLE SIMON
•4 REM     SOLUTION BY
•5 REM      CHARLES KLUEPFEL
•6 REM ==================================
•10 A=RND(-TI):DO:A$="":DO:DO:A$=A$+CHR$(
 48+10*RND(1)):LOOP UNTIL LEN(A$)>2:PRINT
 "[CLEAR]"A$:SLEEP1:INPUT"[CLEAR]";B$:LOO
 P WHILEA$=B$:PRINT"NO,IT IS":PRINTA$:GET
 KEYA$:LOOP
```

Charles said that this illustrates the power of BASIC 7.0. Indeed it does. You may better understand the logic of the program if you rewrite it, indented with one statement per line.

Hopefully you will enjoy improving your memory with these simple Simon solutions. Have fun working on this month's problems. Keep those solutions and suggestions coming. □

Congratulations to the following readers who have not been mentioned already this month:

Necah Buyukdura (Ankara, Turkey)
Harlan Clussman (Wausau, WI)
Oren Dalton (El Paso, TX)
Tameem Hallak
Terry Jernigan (Raleigh, NC)
Larry Louks (Freeport, TX)
Ron McManus (Lenore, ID)
Steve Morrison (Port Hueneme, CA)
Craig Morse (Minot, ND)
Jerry Nichols
Federico Oste (Loreto, Italy)
Myong Paek (Portland, OR)
Donald Pellegrini (Arroyo Grande, CA)
Robert Rispoli (Ridge, NY)
Mark Roschke
Andrew Rosenthal (Flushing, NY)

PROGRAM LISTINGS

Attention new *Ahoy!* readers! You must read the following information very carefully prior to typing in programs listed in *Ahoy!* Certain Commodore characters, commands, and strings of characters and commands will appear in a special format. Follow the instructions and listings guide on this page.

On the following pages you'll find several programs that you can enter on your Commodore computer. But before doing so, read this entire page carefully.

To insure clear reproductions, *Ahoy!'s* program listings are generated on a daisy wheel printer, incapable of printing the commands and graphic characters used in Commodore programs. These are therefore represented by various codes enclosed in brackets [ ]. For example: the SHIFT CLR/HOME command is represented onscreen by a heart . The code we use in our listings is [CLEAR]. The chart below lists all such codes which you'll encounter in our listings, except for one other special case.

The other special case is the COMMODORE and SHIFT characters. On the front of most keys are two symbols. The symbol on the left is obtained by pressing that key while holding down the COMMODORE key; the symbol on the right, by pressing that key while holding down the SHIFT key. COMMODORE and SHIFT characters are represented in our listings by a lower-case "s" or "c" followed by the symbol of the key you must hit. COMMODORE J, for example, is represented by [c J], and SHIFT J by [s J].

Additionally, any character that occurs more than two times in a row will be displayed by a coded listing. For example, [3 "[LEFT]"] would be 3 CuRSoR left commands in a row, [5 "[s EP]"] would be 5 SHIFTed English Pounds, and so on. Multiple blank spaces will be noted in similar fashion: e.g., 22 spaces as [22 " "].

Sometimes you'll find a program line that's too long for the computer to accept (C-64 lines are a maximum of 80 characters, or 2 screen lines long; C-128 lines, a maximum of 160 characters, 2 or 4 screen lines in 40 or 80 columns respectively). To enter these lines, refer to the *BASIC Command Abbreviations Appendix* in your User Manual.

On the next page you'll find our *Bug Repellent* programs for the C-128 and C-64. The version for your machine will help you proofread programs after typing them. (Please note: the *Bug Repellent* line codes that follow each program line, in the whited-out area, should *not* be typed in. See instructions preceding each program.)

On the second page following you will find *Flankspeed*, our ML entry program, and instructions on its use.

Call *Ahoy!* at 212-239-6089 with any problems (if busy or no answer after three rings, call 212-239-0855).

WHEN YOU SEE	IT MEANS	YOU TYPE	YOU WILL SEE
[CLEAR]	Screen Clear	SHIFT CLR/HOME	
[HOME]	Home	CLR/HOME	
[UP]	Cursor Up	SHIFT ↑ CRSR ↓	
[DOWN]	Cursor Down	↑ CRSR ↓	
[LEFT]	Cursor Left	SHIFT ← CRSR →	
[RIGHT]	Cursor Right	← CRSR →	
[SS]	Shifted Space	SHIFT Space	
[INSERT]	Insert	SHIFT INST/DEL	
[DEL]	Delete	INST/DEL	
[RVSON]	Reverse On	CNTRL 9	
[RVSOFF]	Reverse Off	CNTRL 0	
[UPARROW]	Up Arrow	↑	↑
[BACKARROW]	Back Arrow	←	←
[PI]	PI	π	π
[EP]	English Pound	£	£
[BLACK]	Black	CNTRL 1	
[WHITE]	White	CNTRL 2	
[RED]	Red	CNTRL 3	
[CYAN]	Cyan	CNTRL 4	
[PURPLE]	Purple	CNTRL 5	
[GREEN]	Green	CNTRL 6	
[BLUE]	Blue	CNTRL 7	
[YELLOW]	Yellow	CNTRL 8	
[F1]	Function 1	F1	
[F2]	Function 2	SHIFT F1	
[F3]	Function 3	F3	
[F4]	Function 4	SHIFT F3	
[F5]	Function 5	F5	
[F6]	Function 6	SHIFT F5	
[F7]	Function 7	F7	
[F8]	Function 8	SHIFT F7	

BUG REPELLENT FOR THE 64 & 128 By BUCK CHILDRESS

Please note: the *Bug Repellent* programs listed here are for *Ahoy!* programs published from the May 1987 issue onward! For older programs, use the older version.

Type in, save, and run *Bug Repellent*. You'll be asked if you want automatic saves to take place. If so, you're prompted for the device, DISK (D) or TAPE (T). You then pick a starting file number, 0 through 99. Next, you enter a name, up to 14 characters long. At this point, *Bug Repellent* verifies your entries and gives you a chance to change them if you want. If no changes are needed, *Bug Repellent* activates itself. (Pressing RETURN without answering the prompts defaults to disk drive and begins your files with "00BACKUP".)

Type NEW and begin entering an *Ahoy!* program. As you enter program lines and press RETURN, a *Bug Repellent* code appears at the top of your screen. If it doesn't match the code in the program listing, an error exists. Correct the line and the codes will match.

If used, automatic saves take place every 15 minutes. When the RETURN key is pressed on a program line, the screen changes color to let you know that a save will begin in about three seconds. You may cancel the save by pressing the RUN STOP key. The file number increments after each save. It resets to 00 if 99 is surpassed. After saving, or cancelling, the screen returns to its original color and the timer resets for 15 minutes.

When you've finished using *Bug Repellent*, deactivate it by typing SYS 49152 [RETURN] for the Commodore 64 or SYS 4864 [RETURN] for the Commodore 128.

C-64 BUG REPELLENT

```
·10 PRINTCHR$(147)"LOADING AND CHECKING THE DATA[3"."]":J
=49152
·20 FORB=0TO11:READA:IFA<0ORA>255THEN40
·30 POKEJ+B,A:X=X+A:NEXTB:READA:IFA=XTHEN50
·40 PRINT:PRINT"ERROR IN DATA LINE:"PEEK(64)*256+PEEK(63)
:END
·50 X=0:J=J+12:IFJ<49456THEN20
·60 POKE198,0:POKE49456,0:A$="Y":B$=A$:C$="D":D$="DISK":D
=8:PRINTCHR$(147)
·70 INPUT"DO YOU WANT AUTOMATIC SAVES (Y/N)";A$:PRINT:IFA
$="Y"THEN90
·80 PRINT"NO AUTOMATIC SAVES[3"."]":GOTO150
·90 POKE49456,1:INPUT"DISK OR TAPE (D/T)";C$:IFC$<>"D"THE
ND=1:D$="TAPE"
·100 POKE49457,D:D$=D$+" DRIVE":PRINT:INPUT"FILE NUMBER (
0-99)";N
·110 N$=RIGHT$(STR$(N),2):IFN<10THENN$=CHR$(48)+CHR$(N+48
)
·120 F$="BACKUP":PRINT:INPUT"FILENAME";F$:F$=N$+LEFT$(F$,
14):L=LEN(F$)
·130 POKE49458,L:FORJ=1TOL:POKE49458+J,ASC(MID$(F$,J,1)):
NEXTJ:PRINT
·140 PRINT"SAVING DEVICE ** "D$:PRINT"STARTING WITH ** "F
$
·150 PRINT:INPUT"IS THIS CORRECT (Y/N)";B$:IFB$<>"Y"THEN6
0
·160 POKE770,131:POKE771,164:SYS49152:END
·170 DATA169,79,32,210,255,162,38,160,192,204,3,3,1507
·180 DATA208,10,162,131,160,164,169,70,32,210,255,44,1615
·190 DATA169,78,32,210,255,142,2,3,140,3,3,76,1113
·200 DATA36,193,32,96,165,134,122,132,123,32,115,0,1180
·210 DATA170,240,243,162,255,134,58,144,3,76,150,164,1799
·220 DATA32,107,169,32,121,165,173,0,2,240,5,169,1215
·230 DATA79,141,2,3,76,162,164,169,0,133,2,133,1064
·240 DATA251,133,252,133,254,24,101,20,69,254,230,254,197
5
·250 DATA24,101,21,69,254,170,230,254,164,252,185,0,1724
·260 DATA2,133,253,201,34,208,6,165,2,73,255,133,1465
·270 DATA2,201,32,208,4,165,2,240,8,138,24,101,1125
·280 DATA253,69,254,170,44,198,254,230,252,164,253,208,23
49
·290 DATA213,138,41,240,74,74,74,74,24,105,129,141,1327
·300 DATA44,193,138,41,15,24,105,129,141,45,193,162,1230
·310 DATA0,189,43,193,240,12,157,0,4,173,134,2,1147
·320 DATA157,0,216,232,208,239,169,38,141,2,3,173,1578
·330 DATA48,193,240,23,165,161,201,212,176,4,165,160,1748
·340 DATA240,13,238,32,208,160,0,32,225,255,208,6,1617
·350 DATA32,33,193,76,38,192,232,208,242,200,208,239,1893
·360 DATA32,68,229,169,0,168,174,49,193,32,186,255,1555
·370 DATA173,50,193,162,51,160,193,32,189,255,169,43,1670
·380 DATA166,45,164,46,32,216,255,162,1,189,51,193,1520
·390 DATA168,200,152,201,58,144,2,169,48,157,51,193,1543
·400 DATA201,48,208,3,202,16,234,32,33,193,76,116,1362
·410 DATA164,206,32,208,169,0,170,168,76,219,255,160,1827
·420 DATA1,1,160,0,0,65,72,79,89,33,0,0,500
```

C-128 BUG REPELLENT

```
·10 PRINTCHR$(147)"LOADING AND CHECKING THE DATA[3"."]":J
=4864
·20 FORB=0TO11:READA:IFA<0ORA>255THEN40
·30 POKEJ+B,A:X=X+A:NEXTB:READA:IFA=XTHEN50
·40 PRINT:PRINT"ERROR IN DATA LINE:"PEEK(66)*256+PEEK(65)
:END
·50 X=0:J=J+12:IFJ<5213THEN20
·60 POKE208,0:POKE5213,0:A$="Y":B$=A$:C$="D":D$="DISK":D=
8:PRINTCHR$(147)
·70 INPUT"DO YOU WANT AUTOMATIC SAVES (Y/N)";A$:PRINT:IFA
$="Y"THEN90
·80 PRINT"NO AUTOMATIC SAVES[3"."]":GOTO150
·90 POKE5213,1:INPUT"DISK OR TAPE (D/T)";C$:IFC$<>"D"THEN
D=1:D$="TAPE"
·100 POKE5214,D:D$=D$+" DRIVE":PRINT:INPUT"FILE NUMBER (0
-99)";N
·110 N$=RIGHT$(STR$(N),2):IFN<10THENN$=CHR$(48)+CHR$(N+48
)
·120 F$="BACKUP":PRINT:INPUT"FILENAME";F$:F$=N$+LEFT$(F$,
14):L=LEN(F$)
·130 POKE5215,L:FORJ=1TOL:POKE5215+J,ASC(MID$(F$,J,1)):NE
XTJ:PRINT
·140 PRINT"SAVING DEVICE ** "D$:PRINT"STARTING WITH ** "F
$
·150 PRINT:INPUT"IS THIS CORRECT (Y/N)";B$:IFB$<>"Y"THEN6
0
·160 POKE770,198:POKE771,77:SYS4864:END
·170 DATA32,58,20,169,41,162,19,236,3,3,208,4,955
·180 DATA169,198,162,77,141,2,3,142,3,3,224,19,1143
·190 DATA208,7,32,125,255,79,78,0,96,32,125,255,1292
·200 DATA79,70,70,0,96,162,0,134,251,189,0,2,1053
·210 DATA240,19,201,48,144,9,201,58,176,5,133,251,1485
·220 DATA232,208,238,134,252,165,251,208,3,76,198,77,2042
·230 DATA169,0,166,235,164,236,133,253,133,254,142,47,193
2
·240 DATA20,140,48,20,24,101,22,69,254,230,254,24,1206
·250 DATA101,23,69,254,170,230,254,164,252,185,0,2,1704
·260 DATA133,251,201,34,208,6,165,253,73,255,133,253,1965
·270 DATA201,32,208,4,165,253,240,8,138,24,101,251,1625
·280 DATA69,254,170,44,198,254,230,252,164,251,208,213,23
07
·290 DATA138,41,240,74,74,74,74,24,105,65,141,88,1138
·300 DATA20,138,41,15,24,105,65,141,89,20,32,79,769
·310 DATA20,189,85,20,240,6,32,210,255,232,208,245,1742
·320 DATA174,47,20,172,48,20,24,32,240,255,173,93,1298
·330 DATA20,240,27,165,161,201,212,176,4,165,160,240,1771
·340 DATA17,32,65,20,238,32,208,238,1,214,32,225,1322
·350 DATA255,208,6,32,49,20,76,198,77,232,208,242,1603
·360 DATA200,208,239,32,66,193,173,95,20,162,96,160,1644
·370 DATA20,32,189,255,169,0,170,32,104,255,169,0,1395
·380 DATA174,94,20,168,32,186,255,169,45,174,16,18,1351
·390 DATA172,17,18,32,216,255,162,1,189,96,20,168,1346
·400 DATA200,152,201,58,144,2,169,48,157,96,20,201,1448
·410 DATA48,208,3,202,16,234,32,49,20,141,0,2,955
·420 DATA76,183,77,58,59,32,65,20,206,32,208,206,1222
·430 DATA1,214,169,0,170,168,76,219,255,32,79,20,1403
·440 DATA169,26,141,0,214,173,0,214,16,251,96,162,1462
·450 DATA0,142,0,255,96,19,18,32,32,32,32,146,804
·460 DATA0,1,0,0,65,72,79,89,33,0,0,0,339
```

FLANKSPEED FOR THE C-64 By GORDON F. WHEAT

Flankspeed will allow you to enter machine language *Ahoy!* programs without any mistakes. Once you have typed the program in, save it for future use. While entering an ML program with *Flankspeed* there is no need to enter spaces or hit the carriage return. This is all done automatically. If you make an error in a line a bell will ring and you will be asked to enter it again. To LOAD in a program Saved with *Flankspeed* use LOAD "name",1,1 for tape, or LOAD "name",8,1 for disk. The function keys may be used after the starting and ending addresses have been entered.

f1–SAVEs what you have entered so far.

f3–LOADs in a program worked on previously.

f5–To continue on a line you stopped on after LOADing in the previous saved work.

f7–Scans through the program to locate a particular line, or to find out where you stopped the last time you entered the program. It temporarily freezes the output as well.

```
100 POKE53280,12:POKE53281,11                                   OP
105 PRINT"[CLEAR][c 8][RVSON][15" "]FLANKSPEED[15" "]";          FP
110 PRINT"[RVSON][5" "]MISTAKEPROOF ML ENTRY PROGRAM[6" "
 ]"                                                              JP
115 PRINT"[RVSON][9" "]CREATED BY G. F. WHEAT[9" "]"             FA
120 PRINT"[RVSON][3" "]COPR. 1987, ION INTERNATIONAL INC.
 [3" "]"                                                         AJ
125 FORA=54272TO54296:POKEA,0:NEXT                               ND
130 POKE54272,4:POKE54273,48:POKE54277,0:POKE54278,249:PO
 KE54296,15                                                      NP
135 FORA=680TO699:READB:POKEA,B:NEXT                             FL
140 DATA169,251,166,253,164,254,32,216,255,96                    FF
145 DATA169,0,166,251,164,252,32,213,255,96                      EK
150 B$="STARTING ADDRESS IN HEX":GOSUB430:AD=B:SR=B              KP
155 GOSUB480:IFB=0THEN150                                        OE
160 POKE251,T(4)+T(3)*16:POKE252,T(2)+T(1)*16                    AM
165 B$="ENDING ADDRESS IN HEX":GOSUB430:EN=B                     PE
170 GOSUB470:IFB=0THEN150                                        PG
175 POKE254,T(2)+T(1)*16:B=T(4)+1+T(3)*16                        GM
180 IFB>255THENB=B-255:POKE254,PEEK(254)+1                       HG
185 POKE253,B:PRINT                                              EC
190 REM  GET HEX LINE                                            ED
195 GOSUB495:PRINT": [c P][LEFT]";:FORA=0TO8                     KD
200 FORB=0TO1:GOTO250                                            IH
205 NEXTB                                                        IJ
210 A%(A)=T(1)+T(0)*16:IFAD+A-1=ENTHEN340                        FA
215 PRINT" [c P][LEFT]";                                         EG
220 NEXTA:T=AD-(INT(AD/256)*256):PRINT" "                        II
225 FORA=0TO7:T=T+A%(A):IFT>255THENT=T-255                       GL
230 NEXT                                                         GI
235 IFA%(8)<>TTHENGOSUB375:GOTO195                               FL
240 FORA=0TO7:POKEAD+A,A%(A):NEXT:AD=AD+8:GOTO195                IM
245 REM    GET HEX INPUT                                         PA
250 GETA$:IFA$=""THEN250                                         GA
255 IFA$=CHR$(20)THEN305                                         GO
260 IFA$=CHR$(133)THEN535                                        LM
265 IFA$=CHR$(134)THEN560                                        IG
270 IFA$=CHR$(135)THENPRINT" ":GOTO620                           HO
275 IFA$=CHR$(136)THENPRINT" ":GOTO635                           HE
280 IFA$>"@"ANDA$<"G"THENT(B)=ASC(A$)-55:GOTO295                 MI
285 IFA$>"/"ANDA$<":"THENT(B)=ASC(A$)-48:GOTO295                 DJ
290 GOSUB415:GOTO250                                             JA
295 PRINTA$"[c P][LEFT]";                                        PK
300 GOTO205                                                      FA
305 IFA>0THEN320                                                 BI
310 A=-1:IFB=1THEN330                                            BB
315 GOTO220                                                      FA
320 IFB=0THENPRINTCHR$(20);CHR$(20);:A=A-1                       BF
325 A=A-1                                                        FK
330 PRINTCHR$(20);:GOTO220                                       PH
335 REM LAST LINE                                                CP
340 PRINT" ":T=AD-(INT(AD/256)*256)                              KH
345 FORB=0TOA-1:T=T+A%(B):IFT>255THENT=T-255                     OD
350 NEXT                                                         OB
355 IFA%(A)<>TTHENGOSUB375:GOTO195                               LH
360 FORB=0TOA-1:POKEAD+B,A%(B):NEXT                              BO
365 PRINT:PRINT"YOU ARE FINISHED!":GOTO535                       MB
370 REM BELL AND ERROR MESSAGES                                  LM
375 PRINT:PRINT"LINE ENTERED INCORRECTLY":PRINT:GOTO415          JK
380 PRINT:PRINT"INPUT A 4 DIGIT HEX VALUE!":GOTO415              PG
385 PRINT:PRINT"ENDING IS LESS THAN STARTING!":B=0:GOTO41
 5                                                               IK
390 PRINT:PRINT"ADDRESS NOT WITHIN SPECIFIED RANGE!":B=0:
 GOTO415                                                         HK
395 PRINT:PRINT"NOT ZERO PAGE OR ROM!":B=0:GOTO415               DM
400 PRINT"?ERROR IN SAVE":GOTO415                                JK
405 PRINT"?ERROR IN LOAD":GOTO415                                IO
410 PRINT:PRINT:PRINT"END OF ML AREA":PRINT                      JO
415 POKE54276,17:POKE54276,16:RETURN                             BF
420 OPEN15,8,15:INPUT#15,A,A$:CLOSE15:PRINTA$:RETURN             DH
425 REM GET FOUR DIGIT HEX                                       IM
430 PRINT:PRINTB$;:INPUTT$                                       OL
435 IFLEN(T$)<>4THENGOSUB380:GOTO430                             JD
440 FORA=1TO4:A$=MID$(T$,A,1):GOSUB450:IFT(A)=16THENGOSUB
 380:GOTO430                                                     AK
445 NEXT:B=(T(1)*4096)+(T(2)*256)+(T(3)*16)+T(4):RETURN          KB
450 IFA$>"@"ANDA$<"G"THENT(A)=ASC(A$)-55:RETURN                  GM
455 IFA$>"/"ANDA$<":"THENT(A)=ASC(A$)-48:RETURN                  NJ
460 T(A)=16:RETURN                                               IC
465 REM ADDRESS CHECK                                            OL
470 IFAD>ENTHEN385                                               HO
475 IFB<SRORB>ENTHEN390                                          LE
480 IFB<256OR(B>40960ANDB<49152)ORB>53247THEN395                 OB
485 RETURN                                                       HE
490 REM ADDRESS TO HEX                                           PM
495 AC=AD:A=4096:GOSUB520                                        AP
500 A=256:GOSUB520                                               NF
505 A=16:GOSUB520                                                LG
510 A=1:GOSUB520                                                 HE
515 RETURN                                                       JD
520 T=INT(AC/A):IFT>9THENA$=CHR$(T+55):GOTO530                   OC
525 A$=CHR$(T+48)                                                JI
530 PRINTA$;:AC=AC-A*T:RETURN                                    AA
535 A$="**SAVE**":GOSUB585                                       IC
540 OPEN1,T,1,A$:SYS680:CLOSE1                                   AB
545 IFST=0THENEND                                                FB
550 GOSUB400:IFT=8THENGOSUB420                                   PM
555 GOTO535                                                      FI
560 A$="**LOAD**":GOSUB585                                       PE
565 OPEN1,T,0,A$:SYS690:CLOSE1                                   PO
570 IFST=64THEN195                                               OI
575 GOSUB405:IFT=8THENGOSUB420                                   CO
580 GOTO560                                                      GN
585 PRINT" ":PRINTTAB(14)A$                                      KA
590 PRINT:A$="":INPUT"FILENAME";A$                               IO
595 IFA$=""THEN590                                               HK
600 PRINT:PRINT"TAPE OR DISK?":PRINT                             HL
605 GETB$:T=1:IFB$="D"THENT=8:A$="@0:"+A$:RETURN                 NP
610 IFB$<>"T"THEN605                                             KO
615 RETURN                                                       PH
620 B$="CONTINUE FROM ADDRESS":GOSUB430:AD=B                     DD
625 GOSUB475:IFB=0THEN620                                        NK
630 PRINT:GOTO195                                                HN
635 B$="BEGIN SCAN AT ADDRESS":GOSUB430:AD=B                     FK
640 GOSUB475:IFB=0THEN635                                        LN
645 PRINT:GOTO670                                                HI
650 FORB=0TO7:AC=PEEK(AD+B):GOSUB505:IFAD+B=ENTHENAD=SR:G
 OSUB410:GOTO195                                                 LM
655 PRINT" ";:NEXTB                                              LE
660 PRINT:AD=AD+8                                                CD
665 GETB$:IFB$=CHR$(136)THEN195                                  JD
670 GOSUB495:PRINT": ";:GOTO650                                  KE
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 61 and 62 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

VARI-SCAN
FROM PAGE 30

C-64 VERSION

```
·10 REM *** VARI-SCAN 64 *** BUCK CHILDRE
 SS ***                                    EB
·20 REM *** P.O. BOX 13575 SALEM, OR 9730
 9 ***                                     OA
·30 PRINTCHR$(147)"LOADING AND CHECKING D
 ATA LINE:":J=52000                        BM
·40 FORB=0TO11:READA                       MB
·50 IFB=0THENL=PEEK(64)*256+PEEK(63):PRIN
 TCHR$(19)TAB(31)L:PRINT                   PF
·60 IFA<0ORA>255THEN80                     FK
·70 POKEJ+B,A:X=X+A:D=D+1:NEXTB:READA:IFA
 =XTHEN90                                  PL
·80 PRINT"ERROR IN DATA LINE:"L:END        MG
·90 X=0:J=J+12:IFD<564THEN40               OO
·100 PRINT"THE DATA IS OK[3"."]":PRINT     FB
·110 PRINT"SYS 52[3"0"] TO ACTIVATE[3"."]
 ":END                                     AH
·120 DATA104,104,169,79,32,210,255,169,78
 ,162,57,160,1579                          LA
·130 DATA203,142,0,3,140,1,3,32,210,255,7
 6,179,1244                                BP
·140 DATA204,173,0,2,201,95,208,17,169,79
 ,32,210,1390                              JC
·150 DATA255,169,70,32,210,255,162,139,16
 0,227,76,45,1800                          GN
·160 DATA203,201,65,144,4,201,91,144,3,76
 ,139,227,1498                             HP
·170 DATA32,96,166,162,0,160,0,140,85,205
 ,140,83,1269                              OJ
·180 DATA205,189,1,2,240,47,48,198,201,32
 ,208,3,1374                               LP
·190 DATA232,208,242,72,173,85,205,208,10
 ,192,0,240,1867                           AM
·200 DATA6,104,201,48,176,238,72,104,153,
 1,2,201,1306                              FP
·210 DATA48,176,10,141,85,205,201,40,208,
 3,141,83,1341                             NE
·220 DATA205,232,200,208,204,153,1,2,165,
 43,166,44,1623                            DA
·230 DATA133,251,134,252,134,254,160,0,17
 7,43,133,253,1924                         BO
·240 DATA152,145,45,200,208,251,141,90,20
 5,141,92,205,1875                         EP
·250 DATA32,2,205,160,2,177,251,200,209,2
 51,208,7,1704                             HN
·260 DATA201,0,208,3,76,167,204,170,177,2
 51,141,88,1686                            HA
·270 DATA205,142,89,205,160,3,32,2,205,14
 1,91,205,1480                             IA
·280 DATA173,92,205,240,8,162,0,142,92,20
 5,32,187,1538                             AL
·290 DATA204,200,177,251,208,3,76,135,204
 ,201,131,240,2030                         AL
·300 DATA4,201,143,208,3,76,138,204,201,3
 4,208,7,1427                              IK
·310 DATA174,91,205,240,212,208,207,174,9
 1,205,208,208,2223                        GE
·320 DATA201,65,144,22,201,91,176,200,141
 ,92,205,174,1712                          PJ
·330 DATA84,205,208,55,174,87,205,224,2,1
 44,51,76,1515                             EP
·340 DATA237,203,174,92,205,240,190,201,5
 8,176,173,201,2150                        BG
·350 DATA48,176,228,201,32,240,178,201,36
 ,240,21,201,1802                          EP
·360 DATA37,240,17,201,40,240,13,201,41,2
 40,9,201,1480                             BC
·370 DATA44,208,29,174,83,205,240,24,141,
 84,205,174,1611                           KC
·380 DATA87,205,157,93,205,238,87,205,201
 ,41,208,11,1738                           BF
·390 DATA173,83,205,208,3,206,87,205,76,2
 24,203,201,1874                           CG
·400 DATA40,208,17,232,142,86,205,173,83,
 205,208,8,1607                            AP
·410 DATA169,0,141,92,205,141,87,205,76,2
 37,203,32,1588                            BA
·420 DATA187,204,32,5,205,166,253,165,254
 ,240,20,134,1865                          DN
·430 DATA251,133,252,160,0,177,253,72,200
 ,177,253,133,2061                         HI
·440 DATA254,104,133,253,76,191,203,173,9
 0,205,208,7,1897                          KL
·450 DATA169,47,160,205,32,30,171,169,0,1
 41,0,2,1126                               GP
·460 DATA76,116,164,162,0,189,0,2,240,18,
 201,92,1260                               GO
·470 DATA208,6,236,86,205,76,215,204,221,
 93,205,208,1963                           JJ
·480 DATA49,232,208,233,236,87,205,208,41
 ,104,104,173,1880                         FI
·490 DATA90,205,208,7,169,21,160,205,32,3
 0,171,169,1467                            GK
·500 DATA76,160,205,32,30,171,173,88,205,
 174,89,205,1608                           LG
·510 DATA32,205,189,169,13,32,210,255,141
 ,90,205,76,1617                           NJ
·520 DATA138,204,169,0,170,157,93,205,232
 ,208,250,141,1967                         IE
·530 DATA84,205,141,86,205,141,87,205,96,
 13,18,86,1367                             DK
·540 DATA65,82,73,65,66,76,69,32,73,83,32
 ,73,789                                   IB
·550 DATA78,32,85,83,69,46,46,46,13,13,0,
 13,524                                    JA
```

```
•560 DATA18,86,65,82,73,65,66,76,69,32,73
 ,83,788                                   MI
•570 DATA32,78,79,84,32,73,78,32,85,83,69
 ,46,771                                   GN
•580 DATA46,46,13,0,76,73,78,69,58,32,0,2
 20,711                                    CO
```

C-128 VERSION

```
•10 REM *** VARI-SCAN 128 *** BUCK CHILDR
 ESS ***                                   ED
•20 REM *** P.O. BOX 13575 SALEM, OR 9730
 9 ***                                     OA
•30 PRINTCHR$(147)"LOADING AND CHECKING D
 ATA LINE:":J=4864                         AD
•40 FORB=0TO11:READA                       MB
•50 IFB=0THENL=PEEK(66)*256+PEEK(65):PRIN
 TCHR$(19)TAB(31)L:PRINT                   LB
•60 IFA<0ORA>255THEN80                     FK
•70 POKEJ+B,A:X=X+A:D=D+1:NEXTB:READA:IFA
 =XTHEN90                                  PL
•80 PRINT"ERROR IN DATA LINE:"L:END        MG
•90 X=0:J=J+12:IFD<636THEN40               OO
•100 PRINT"THE DATA IS OK[3"."]":PRINT     FB
•110 PRINT"SYS 4864 TO ACTIVATE[3"."]":EN
 D                                         AJ
•120 DATA104,104,32,129,146,79,78,0,169,2
 9,162,19,1051                             ND
•130 DATA141,2,3,142,3,3,169,69,141,0,3,1
 42,818                                    PI
•140 DATA1,3,76,236,20,169,0,162,19,133,2
 51,141,1211                               AN
•150 DATA0,255,134,252,169,251,141,185,2,
 162,1,160,1712                            EJ
•160 DATA0,177,251,32,119,255,230,251,208
 ,238,230,252,2243                         IN
•170 DATA165,252,201,22,144,230,76,198,77
 ,173,0,2,1540                             IH
•180 DATA201,95,208,22,32,129,146,79,70,7
 0,0,169,1221                              KK
•190 DATA198,162,77,141,2,3,142,3,3,169,6
 3,76,1039                                 GP
•200 DATA20,19,201,65,144,4,201,91,144,3,
 76,63,1031                                PL
•210 DATA77,32,250,81,162,0,160,0,140,120
 ,21,140,1183                              KD
•220 DATA118,21,189,1,2,240,47,48,153,201
 ,32,208,1260                              JG
•230 DATA3,232,208,242,72,173,120,21,208,
 10,192,0,1481                             DE
•240 DATA240,6,104,201,48,176,238,72,104,
 153,1,2,1345                              PP
•250 DATA201,48,176,10,141,120,21,201,40,
 208,3,141,1310                            HN
•260 DATA118,21,232,200,208,204,153,1,2,1
 65,45,166,1515                            AF
•270 DATA46,133,251,134,252,134,254,160,0
 ,177,45,133,1719                          IB
•280 DATA253,173,16,18,174,17,18,133,65,1
 34,66,152,1219                            GO
•290 DATA145,65,200,208,251,141,125,21,14
 1,127,21,32,1477                          HB
•300 DATA92,21,160,2,177,251,200,209,251,
 208,7,201,1779                            AO
•310 DATA0,208,3,76,194,20,170,177,251,14
 1,123,21,1384                             CK
•320 DATA142,124,21,160,3,32,92,21,141,12
 6,21,173,1056                             HJ
•330 DATA127,21,240,8,162,0,142,127,21,32
 ,247,20,1147                              JP
•340 DATA200,177,251,208,3,76,162,20,201,
 131,240,4,1673                            JC
•350 DATA201,143,208,3,76,165,20,201,34,2
 08,7,174,1440                             DJ
•360 DATA126,21,240,212,208,207,174,126,2
 1,208,208,201,1952                        GL
•370 DATA65,144,22,201,91,176,200,141,127
 ,21,174,119,1481                          HH
•380 DATA21,208,55,174,122,21,224,2,144,5
 1,76,8,1106                               JO
•390 DATA20,174,127,21,240,190,201,58,176
 ,173,201,48,1629                          HB
•400 DATA176,228,201,32,240,178,201,36,24
 0,21,201,37,1791                          EI
•410 DATA240,17,201,40,240,13,201,41,240,
 9,201,44,1487                             MN
•420 DATA208,29,174,118,21,240,24,141,119
 ,21,174,122,1391                          FJ
•430 DATA21,157,128,21,238,122,21,201,41,
 208,11,173,1342                           HP
•440 DATA118,21,208,3,206,122,21,76,251,1
 9,201,40,1286                             CJ
•450 DATA208,17,232,142,121,21,173,118,21
 ,208,8,169,1438                           JM
•460 DATA0,141,127,21,141,122,21,76,8,20,
 32,247,956                                JF
•470 DATA20,32,95,21,166,253,165,254,240,
 20,134,251,1651                           AA
•480 DATA133,252,160,0,177,253,72,200,177
 ,253,133,254,2064                         EG
•490 DATA104,133,253,76,218,19,173,125,21
 ,208,37,169,1536                          JF
•500 DATA0,141,0,255,32,129,146,13,18,86,
 65,82,967                                 OI
•510 DATA73,65,66,76,69,32,73,83,32,78,79
 ,84,810                                   FJ
•520 DATA32,73,78,32,85,83,69,46,46,46,13
 ,0,603                                    AN
•530 DATA169,0,141,0,2,141,0,255,76,55,77
 ,162,1078                                 FJ
•540 DATA0,189,0,2,240,18,201,92,208,6,23
 6,121,1313                                DK
•550 DATA21,76,19,21,221,128,21,208,79,23
 2,208,233,1467                            HL
•560 DATA236,122,21,208,71,104,104,169,0,
 141,0,255,1431                            FM
```

```
•570 DATA173,125,21,208,29,32,129,146,13,
18,86,65,1045                              HO
•580 DATA82,73,65,66,76,69,32,73,83,32,73
,78,802                                    JJ
•590 DATA32,85,83,69,46,46,46,13,13,0,32,
129,594                                    KF
•600 DATA146,76,73,78,69,58,32,0,173,123,
21,174,1023                                MO
•610 DATA124,21,32,50,142,169,13,32,210,2
55,141,125,1314                            KM
•620 DATA21,76,165,20,169,0,170,157,128,2
1,232,208,1367                             IN
•630 DATA250,141,119,21,141,121,21,141,12
2,21,169,63,1330                           KJ
•640 DATA141,0,255,169,0,96,0,0,65,72,83,
0,881                                      HC
```

THERMAL CONNECTION
FROM PAGE 32

C-64 PADDLE ROUTINE

```
1 REM==================================
3 REM    C-64 PADDLE ROUTINE
4 REM      RUPERT REPORT #55
5 REM
6 REM       MONITOR LISTING
8 REM==================================
9  REM DISABLE INTERRUPTS   (LINE 90)
10 REM SAVE CIA DDR        (LINE 100)
11 REM SET BITS 7 & 6 OF CIA
12 REM  #1 AS OUTPUTS       (110-120)
13 REM ENABLE POTS A AND B      (130)
14 REM PUT DELAY COUNT IN X     (140)
15 REM DELAY LOOP           (150-170)
16 REM READ POT X               (180)
17 REM STORE POT X IN 251       (190)
18 REM READ POT Y               (200)
19 REM STORE POT Y IN 252       (210)
20 REM RESTORE CIA DDR          (220)
21 REM ENABLE INTERRUPTS        (230)
22 REM BACK TO BASIC            (240)
23 REM==================================
90  "0C000  78        SEI
100 "0C001  AC 02 DC  LDY $DC02
110 "0C004  A9 C0     LDA #$C0
120 "0C006  8D 02 DC  STA $DC02
130 "0C009  8D 00 DC  STA $DC00
140 "0C00C  AA        TAX
150 "0C00D  EA        NOP
160 "0C00E  CA        DEX
170 "0C00F  10 FC     BPL $C00D
180 "0C011  AD 19 D4  LDA $D419
190 "0C014  85 FB     STA $FB
200 "0C016  AD 1A D4  LDA $D41A
210 "0C019  85 FC     STA $FC
220 "0C01B  8C 02 DC  STY $DC02
230 "0C01E  58        CLI
240 "0C01F  60        RTS
```

THERMAL CONNECTION

```
•1 REM ==================================  AO
•2 REM                                     JD
•3 REM       THERMAL CONNECTION            FF
•4 REM                                     JF
•5 REM       RUPERT REPORT #55             CE
•6 REM                                     JH
•7 REM ==================================  NE
•8 REM   CALIBRATE AND READ THERMISTOR     FH
•9 REM    CONNECTED TO PADDLE INPUT        PL
•10 REM ==================================  JL
•20 DEF FNR(X)=INT(10*X+.5)/10        :REM
 ROUND OFF TO 1 DECIMAL                    JO
•30 DEF FNK2F(T)=1.8*(T-273)+32       :REM
 CONVERT DEG K TO DEG F                    JP
•40 DEF FNF2K(T)=273+(T-32)/1.8       :REM
 CONVERT DEG F TO DEG K                    AE
•50 DEF FNT(R)=1/(1/T0+(1/B)*LOG(R/R0)) :
 REM TEMP IN DEG K                         BJ
•60 GOSUB 3000        :REM GET CALIBRATION
 CONSTANTS                                 NG
•70 PRINT "PUT THERMISTOR AT NEW TEMPERAT
URE"                                       GF
•80 PRINT "PRESS ANY KEY WHEN READY"       KP
•90 GET K$: IF K$="" THEN 90               JA
•95 REM  ------  MAIN LOOP  ------------- FH
•100 R=POT(1)                              LN
•110 T=FNT(R)  :REM TEMP IN DEG K          MM
•120 PRINT "TEMPERATURE :";FNR(FNK2F(T));
 "DEG F  ";                                HA
•130 PRINT FNR(T-273); "DEG C"             OO
•140 GOTO 100                              KB
•1000 REM ---- WAIT FOR SETTLING -------- KF
•1010 PRINT "PRESS A KEY WHEN VALUE HAS S
ETTLED :"                                  EC
•1020 P1=POT(1)                            JB
•1030 PRINT P1;                            FO
•1040 GET K$: IF K$="" THEN 1020           MB
•1050 RETURN                               KM
•2000 REM ---- AVERAGE VALUES ----------  JE
•2010 SUM=0                                LM
•2020 FOR N=1 TO 10                        JL
•2030 P1=POT(1)                            HD
•2040 SUM=SUM+P1: NEXT                     HL
•2050 RN=SUM/10                            FH
•2060 RETURN                               KC
•3000 REM ---- GET CALIB. CONSTANTS ----- EM
•3010 PRINT "[CLEAR]DO YOU KNOW CALIBRATI
ON CONSTANTS (Y/N)": INPUT K$              CJ
•3020 IF K$="N" THEN 3070 :REM CALIBRATE  MH
•3030 INPUT "WHAT IS B"; B                 JN
```

```
·3040 INPUT "WHAT IS T0"; T0                 CM
·3050 INPUT "WHAT IS R0"; R0                 AO
·3060 GOTO 3270  :REM RETURN                 NP
·3070 PRINT "PUT THERMISTOR AT ROOM TEMP"    IK
·3080 INPUT"WHAT IS ROOM TEMP (DEG F)";T1    NC
·3090 GOSUB 1010 :REM DISPLAY RESISTANCE     EM
·3100 GOSUB 2010 :REM RETURN VALUE           LO
·3110 R1=RN                                  AN
·3120 PRINT: PRINT "PUT THERMISTOR IN ICE
 WATER"                                      CI
·3130 PRINT "PRESS ANY KEY WHEN READY"       JF
·3140 GET K$: IF K$="" THEN 3140             OG
·3150 GOSUB 1010                             KJ
·3160 GOSUB 2010                             LC
·3170 R0=RN                                  EK
·3180 T1=FNF2K(T1): T0=273                   FF
·3190 PRINT: PRINT"[38"="]"                  EP
·3200 B=LOG(R1/R0)/(1/T1-1/T0)               DH
·3210 PRINT "RECORD THESE CALIBRATION CON
STANTS :"                                    KH
·3220 PRINT "B =";FNR(B)                     OG
·3230 PRINT "T0 =";T0                        DP
·3240 PRINT "R0 =";R0                        DN
·3250 PRINT"[38"="]"                         ML
·3260 PRINT "THERMISTOR IS NOW CALIBRATED
": PRINT: PRINT                              AA
3270 RETURN                                  GA
4990 REM ################################### HO
4991 REM FOLLOWING LINES FOR C-64 ONLY       NE
4992 REM ################################### FM
4993 REM ADD LINE 11                         JA
4994 REM REPLACE LINES 100, 1020, 2030       LK
4995 REM ADD LINES 5010-49184                GB
4996 REM                                     CE
5000 REM   11 SA=49152: GOSUB 5010           ME
5001 REM  100 SYS SA: R=PEEK(251)            FM
5002 REM 1020 SYS SA: P1=PEEK(251)           MD
5003 REM 2030 SYS SA: P1=PEEK(251)           DG
5004 REM=============================        JC
5010 M=49152: CSUM=4609                      IE
5020 READ B$: IF B$="XX" THEN 5100           NO
5030 B=VAL(B$)                               CM
5040 POKE M,B: M=M+1: CK=CK+B                EE
5050 GOTO 5020                               BJ
5100 IF CK=CSUM THEN RETURN                  EN
5110 PRINT "DATA ERROR.  CHECK TYPING.":
 END                                         OI
49152 DATA 120,172,2,220,169,192,141,2       KB
49160 DATA 220,141,0,220,170,234,202,16      LB
49168 DATA 252,173,25,212,133,251,173,26     MN
49176 DATA 212,133,252,140,2,220,88,96       NP
49184 DATA XX                                BE
```

LUNAR BUGGY
FROM PAGE 15

```
10 POKE53248+21,.:COLOR0,1:COLOR4,1:PRIN
TCHR$(11)CHR$(142)CHR$(147):CHAR,10,5,"[
c 5][s U][4"[s C]"][s I][DOWN][6"[LEFT]"
][s B][4"Z"][s B][DOWN][6"[LEFT]"][s J][
4"[s C]"][s K][s Q][DOWN][s W][DOWN][s W
][DOWN].[DOWN]."                             NP
·20 CHAR,20,12,"[c 8][4"[c @]"][DOWN][5"[
LEFT]"][s N][s F][s F] [s F][s M][DOWN][
7"[LEFT]"][s N][4"[c +]"] [c +][s M][DOW
N][8"[LEFT]"][8"[c U]"]":CHAR,23,10,"(WO
RKING)"                                      BK
·30 FORX=3584TO4096:READA$:POKEX,DEC(A$):
NEXT:FORX=4864TO5054:READA$:POKEX,DEC(A$
):NEXT                                       JE
·40 FORX=.TO63:POKE55*64+X,.:NEXT:POKE55*
64+32,192:POKE55*64+35,192:PRINTCHR$(147
)                                            HE
·50 POKE4593,203:POKE4592,100:POKE4595,0:
POKE4594,30:POKE4602,250:POKE4597,0:POKE
4598,130:POKE4599,56:SYS4864                 EO
·60 B$="":FORX=.TO39:B$=B$+CHR$(32):NEXT:
FORX=1TO8:MOVSPRX,.#.:NEXT                   EA
·70 TEMPO10:D$(.)="01":D$(2)="02":D$(3)="
03":D$(4)="04":D$(5)="04":D$(6)="01"         MJ
·80 FORX=4TO7:POKE2040+X,62:NEXT:FORX=5TO
6:SPRITEX,.,13,.,1,1,.:NEXT:FORX=7TO8:SP
RITEX,.,12,.,1,.,.:NEXT:POKE2040+3,63:PO
KE2041,55:SPRITE2,.,2,.,.,.,.                PI
·90 MOVSPR5,30,152:MOVSPR6,250,152:MOVSPR
7,120,157:MOVSPR8,350,157:FORX=5TO6:MOVS
PRX,270#2:MOVSPRX+2,270#1:NEXT:POKE2040,
56:SPRITE1,.,14,.,.,.,.:MOVSPR1,100,198      EI
·100 POKE2043,60:POKE2042,59:SPRITE3,.,4,
.,.,.,1:MOVSPR3,370,150:SPRITE4,.,13,.,1
,1,.:MOVSPR4,370,198:MOVSPR4,270#3           BC
·110 GOSUB380:PRINTCHR$(27)"M";"[4"[DOWN]
"][RVSON][c 4]"B$B$"[c 5]"B$B$B$"[c 8]"B
$B$B$B$                                      CH
·120 DIMD(8),S(1),M$(2)                      MN
·130 J=.:D=.:G=15:F=127:Z=.:V=1:Q=.:S=.:B
=.:H=5:P=10:I=5:K=25:U=1:T=2:M=3:L=198:S
C=.:N=4:Q=240:VIC=53248:S(.)=315:S(1)=22
5                                            AA
·140 A$="[HOME][5"[DOWN]"][21"[RIGHT]"]":
B$="[8"[RIGHT]"]":M$(.)=" YOU RECOVERED
THE BUGGY! ":M$(1)=" YOU CRASHED IN A CR
ATER! ":M$(2)=" YOU COLLIDED WITH THE SH
IP!"                                         AN
·150 FORX=1TO10:READB$(X):NEXT               AA
·160 FORX=.TO8:D(X)=-1:NEXT:D(.)=90:D(1)=
0:D(3)=90:F$="[5"#"]":H$="[4"#"]":SPRCOL
OR7,3                                        BE
·170 DEFFNJ(X)=(C<H)*T+(C=>H)*-T+(C=P)*T     KA
·180 DEFFNF(U)=-(J>FANDD(X)=>.ANDRSPRITE(
T,.)=.):DEFFNC(X)=(J=H)*-U+((JANDG)=H+U)
*-T+((JAND(H+M))>.)*-M                       IE
·190 POKEVIC+21,241:X=.:DO:PLAYD$(X):PLAY
"T7IG#GGEG#GGE":X=X+1+(X=6)*7:LOOPWHILEJ
OY(2)<127:WINDOW0,0,39,15,1:PRINT"[HOME]
```

```
[HOME]":FORJ=.TO99:X=RND(U)*585:POKE1024
+X,46                                          HA
•200 POKE55296+X,RND(U)*P+U:NEXT               HB
•210 CHAR,1,5,"[GREEN]MILES TO NEXT BASE:
 [c 6][5"0"]  [GREEN]SCORE [c 6][5"0"]"        ON
•220 GOSUB360:B=500:Q=(10-V)*.1:CHAR,15,2
,"[c 7]BASE [WHITE]":PUDEF" ":PRINTUSING
H$;B$(V):PUDEF"0":MOVSPR3,480,160:SPRITE
3,1,V+1:POKE4595,1                             KB
•230 J=BUMP(U):J=BUMP(U)                       BL
•240 REM MAIN LOOP                             NG
•250 DO:J=JOY(T):X=JOY(T)ANDG:IFDTHENMOVS
PRU,+.,+D:C=C+U:D=FNJ(.):ELSEMOVSPRU,+.,
L:IFX=UANDJ<FTHEND=-U:C=.:SOUNDU,F*F,K,.
,F*M,F*M,.:ELSEIFRND(.)>QTHENZ=-(Z=.)          BB
•260 ONFNF(.)GOSUB350:MOVSPRM,S(Z)#V:PRIN
TA$;:PRINTUSINGF$;B;:B=B-U:PRINTB$;:PRIN
TUSINGF$;S:J=BUMP(U):LOOPUNTILB<.ORFNC(.
):ONFNC(.)GOTO300,340,310                      CP
•270 MOVSPRU,+.,L:V=V+U:D=.:IFV<11THEN220      IJ
•280 FORX=1TO8:MOVSPRX,.#.:NEXT:PLAY"V1O1
Q$BV2O2I$BV1O5.W$BV2O3I$BO4I$EV3O3IGI$BV
2O4IGIGV3I$EO3I$BV2IFI$EV3O3IGI$BV2O4IFQ
GV3Q$EO3Q$BV2O4Q$EQ$EV3O3QG":X=.               OF
•290 J=1:DO:PRINT"[HOME][7"[DOWN]"][RVSON
]"TAB(6)M$(X):COLOR5,J:J=J+1+(J=16)*16:L
OOPUNTILJOY(T)>127:RUN                         FC
•300 GOSUB320:X=2:GOTO290                      AB
•310 GOSUB320:X=1:GOTO290                      BC
•320 SYS65418:FORX=1TO8:MOVSPRX,.#.:NEXT:
FORX=1TO3:SPRITEX,.,2,.,.,.,.:MOVSPRX,RS
PPOS(U,.),RSPPOS(U,U):POKE2039+X,61:NEXT JE
•330 SOUND3,4000,120,1,500,500,3:FORX=1TO
3:MOVSPRX,X*45+20#2:SPRITEX,U:NEXT:SLEEP
2:POKEVIC+21,248:RETURN                        CF
•340 POKE2042,61:MOVSPRM,.#.:SOUNDM,F*P,K
*M,T,K*M,K*M,M:SPRITEM,.:MOVSPRM,480,+.:
POKE2042,59:SPRITEM,U:S=S+100*V:GOTO230        FN
•350 SPRITET,.:MOVSPRT,.#.:MOVSPRT,RSPPOS
(U,.),RSPPOS(U,U):MOVSPRT,D(X)#P:SOUNDT,
F*K,K,U,F,F,M:SPRITET,U:RETURN                 BG
•360 SPRITE3,.:SPRITE4,.,2,.,1,1,1:MOVSPR
4,470,176:POKE2043,63:SPRITE4,1:DO:X=RSP
POS(4,.):LOOPUNTILX>470ANDX<480                MA
•370 SPRITE4,.,13,.,1,1,.:MOVSPR4,480,198
:POKE2043,60:SPRITE4,1:RETURN                  EG
•380 PRINT"[HOME]"TAB(7)"[YELLOW]AHOY! MA
GAZINE PRESENTS[DOWN]"                         PH
•390 PRINTTAB(8)"[GREEN][RVSON]  [RVSOFF]
  [RVSON]  [c J] [c G] [c *]  [c G][3" "
][c G][3" "]"                                  JE
•400 PRINTTAB(8)"[RVSON]  [RVSOFF]  [RVSO
N]  [c J] [c G][4" "][c G][c C][c V] [c
G][c C][c V] "                                 KK
•410 PRINTTAB(8)"[RVSON]  [RVSOFF]  [RVSO
N]  [c J] [c G] [RVSOFF][c *][RVSON]  [c
 G] [c G] [c G] [RVSOFF][c *][RVSON] [c
*]"                                            FH
•420 PRINTTAB(8)"[RVSON][3" "][c M][4" "]
[c G] [RVSOFF] [RVSON]  [c G] [c G] [c G
] [RVSOFF] [RVSON]  "                          OJ
•430 PRINTTAB(8)"[c 6][RVSON]  B[3" "]U[3
" "]G[3" "]G[3" "]Y!  "                        OA
•440 PRINTTAB(12)"[YELLOW][DOWN][DOWN]FOR
 THE C-128!":PRINTTAB(4)"[DOWN][c 6](C)1
988 CLEVELAND M. BLAKEMORE":RETURN             CP
•450 REM SPRITE DATA                           CG
•460 DATA 0F,80,,10,47,,23,22,,2E,22,,7F,
FF,80,FF,FF,E0,FF,1,F0,FE,BA,F8,54,BA,55
,FF,1,FF,FF,B9,FD,7F,76                        KI
•470 DATA FC,3E,E7,78,1,DB,80,3,81,C0,7D,
,BE,D6,,6B,92,,49,92,,49,D6,,6B,7C,,3E,,
F,80,,10,47,0                                  FE
•480 DATA 23,22,,2E,22,,7F,FF,80,FF,FF,E0
,FF,1,F0,FE,BA,F8,54,BA,55,FF,1,FF,FF,B9
,FD,7F,76,FC,3E,E7,78,1,DB                     CK
•490 DATA 80,3,81,C0,7D,,BE,C6,,63,9A,,4D
,B2,,59,C6,,63,7C,,3E,,F,80,,10,47,,23,2
2,,2E,22,0                                     OB
•500 DATA 7F,FF,80,FF,FF,E0,FF,1,F0,FE,BA
,F8,54,BA,55,FF,1,FF,FF,B9,FD,7F,76,FC,3
E,E7,78,1,DB,80,3,81,C0,7D,0                   KD
•510 DATA BE,C6,,63,B2,,59,9A,,4D,C6,,63,
7C,,3E,,,AA,,2,14,80,2,14,80,2A,AA,A8,99
,96,66,AA,AA,AA                                HH
•520 DATA 2A,AA,A8,,FF,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,0                                   GK
•530 DATA0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,0                                           AJ
•540 DATA0,,,,,FF,FF,FF,7F,FF,FE,7F,FF,F
8,FF,FF,FC,6F,FF,FE,F,FF,F0,F,FF,30,C,FF
,90,,73,80,,20                                 NE
•550 DATA0,,,,,24,40,9,,,,24,48,2,1,1,20
,,20,8,10,2,80,4A,88,11,38,1,,7D,8,42,90
,0                                             EB
•560 DATA0,28,82,90,82,,,10,8,20,44,1,8,,
28,,81,,24,,10,,24,40,1,,,,4,,,,,,0            LM
•570 DATA0,,,,,,,,,,,,40,,,42,,,C3,,1,C3,
,1,C3,,1,E3,80,3,E7,80,7,FF,80                 DH
•580 DATA 07,FF,C0,7,FF,E0,7,FF,F0,7,FF,F
0,F,FF,F0,F,FF,F8,3F,FF,FC,7F,FF,FC,FF,F
F,FF,,,5,FC,,5,A8,0                            CC
•590 DATA 0F,FC,,A,A8,,F,FC,,8,,,8,,,8,,5
,55,50,15,55,54,55,55,55,66,69,99,55,55,
55,66,69,99                                    GJ
•600 DATA 55,55,55,55,55,55,56,96,95,56,9
6,95,56,96,95,56,96,95,56,96,95,,7             KF
•610 REM INTERRUPT DATA                        NO
•620 DATA 78,A9,D,8D,14,3,A9,13,8D,15,3,5
8,60,AD,F3,11,D0,3,4C,2F,13,AD,,DC,29,F,
85,FD,A9,F,38,E5,FD,C9,4                       GA
•630 DATA D0,3,CE,D6,11,C9,8,D0,3,EE,D6,1
1,AD,D6,11,38,CD,F2,11,B0,9,AD,F2,11,8D,
D6,11,4C,50,13,AD,D6,11,38,CD                  AN
•640 DATA FA,11,90,6,AD,FA,11,8D,D6,11,A2
,7,8E,F4,11,AD,F4,11,A,AA,E8,BD,D6,11,38
```

```
,CD,F0,11,B0,15,AD,F0,11,9D,D6                ID
·650 DATA 11,E0,3,D0,1A,A9,FD,2D,15,D0,8D
,15,D0,4C,88,13,BD,D6,11,38,CD,F1,11,90,
6,AD,F1,11,9D,D6,11,CE,F4,11,30               DF
·660 DATA 3,4C,55,13,CE,F5,11,30,16,AD,F6
,11,8D,F5,11,AE,F7,11,E8,E0,3B,D0,2,A2,3
8,8E,F7,11,8E,F8,7,AD,D8,11,38                HC
·670 DATA CD,FA,11,90,8,A9,FD,2D,15,D0,8D
,15,D0,4C,65,FA                               GM
·680 REM BASES                                LN
·690 DATA I,II,III,IV,V,VI,VII,VIII,IX,X      IO
```

WINDOW DRESSING
FROM PAGE 37

BASIC LOADER

```
·100 REM *** WINDOW DRESSING LOADER ***       NC
·110 SA=4864:REM WILL RELOCATE                LA
·120 CK=0                                     OL
·130 READD:CK=CK+D:IFD=999THEN150             AG
·140 GOTO130                                  KO
·150 IFCK<>30632THENPRINT"ERROR IN DATA":
END                                           IC
·160 RESTORE                                  CM
·170 NA=SA                                    IP
·180 READD:IFD=999THEN230                     AL
·190 IFD=>0THENPOKENA,D:GOTO220               LP
·200 AD=SA+ABS(D):H=AD/256:L=AD-INT(AD/25
6)*256                                        BB
·210 POKENA,L:NA=NA+1:POKENA,H                BO
·220 NA=NA+1:GOTO180                          OA
·230 AD=SA+246:H=AD/256:L=AD-INT(AD/256)*
256                                           LE
·240 POKESA+181,L:POKESA+183,H                NF
·250 PRINT"WINDOW DRESSING INSTALLED"SA"T
O"NA                                          CN
·260 PRINT"SYS"SA"FOR FRAME"                  CG
·270 PRINT"SYS"SA+3"FOR PRINT AT"             EO
·280 PRINT"SYS"SA+6"FOR CRSR RESTORE"         HJ
·290 PRINT"SYS"SA+9"FOR EDITOR SAVE/RECAL
L"                                            GF
·300 END                                      KN
·310 DATA 184,80,26,184,80,86,184,80          MI
·320 DATA 122,184,80,113,176,192,174,221      MF
·330 DATA 29,221,173,192,189,0,0,0            FP
·340 DATA 0,40,80,120,160,32,80,193           IA
·350 DATA 165,248,133,207,32,229,202,162      OE
·360 DATA 255,32,-155,164,229,200,132,206 MC
·370 DATA 32,-155,230,206,164,206,196,228 JI
·380 DATA 240,5,202,202,202,208,240,32        MM
·390 DATA -155,32,84,200,32,-144,165,207      JA
·400 DATA 133,248,230,229,230,230,198,228 JC
·410 DATA 198,231,32,50,202,76,111,199        NH
·420 DATA 72,165,236,141,-22,165,235,141      FM
·430 DATA -23,24,32,24,192,176,7,32           MP
·440 DATA -144,104,240,16,96,104,173,-21      BM
·450 DATA 208,249,76,40,125,201,5,144         AI
·460 DATA 48,176,242,173,-22,133,236,173      MI
·470 DATA 23,19,133,235,76,92,193,32          DD
·480 DATA 134,3,240,222,32,92,121,76          HD
·490 DATA 90,85,32,-170,164,230,200,232       IP
·500 DATA 32,-171,200,196,231,208,248,232 NP
·510 DATA 189,-12,76,12,192,134,195,168       EM
·520 DATA 169,246,162,19,24,121,-24,144       IE
·530 DATA 1,232,164,195,133,195,134,196       CF
·540 DATA 162,26,152,208,22,160,37,136        HJ
·550 DATA 181,224,145,195,202,16,248,162      GB
·560 DATA 13,136,189,84,3,145,195,202         LM
·570 DATA 16,247,96,160,37,136,177,195        LL
·580 DATA 149,224,202,16,248,162,13,136       CN
·590 DATA 177,195,157,84,3,202,16,247         DM
·600 DATA 96,0,999                            HB
```

DEMO

```
·100 REM WINDOW DRESSING DEMO                 GA
·110 GOSUB510                                 JG
·120 COLORBG,15:COLOR4,15                     FA
·130 COLOR5,8:WINDOW0,0,SW-1,4:SYS FR,,,,
,M$;1:SLEEP1:PRINT"[DOWN][RIGHT]AS YOU C
AN SEE[3"."]"                                 OO
·140 SLEEP1:PRINTEQ$"[RIGHT]THESE ROUTINE
S[3"."]":SLEEP1:SYS ED,0,0                    PC
·150 COLOR5,14:WINDOW0,5,SW/2-1,24:SYS FR
,,,,,M$;2:SYS ED,1,0                          KN
·160 SYS ED,0,1:PRINTEQ$"[RIGHT]DRESS UP
YOUR WINDOWS,":SLEEP1                         HG
·170 COLOR5,2:WINDOWSW/2,5,SW-1,24:SYS FR
,,,,,M$;3:SYS ED,2,0                          FO
·180 SYS ED,0,1                               JD
·190 PRINTEQ$"[RIGHT]LET YOUR PROGRAMS[3"
."]":SLEEP1                                   DO
·200 PRINTEQ$"[RIGHT]JUMP FROM WINDOW[3".
"]":SLEEP1                                    HI
·210 SYS ED,1,1:LIST:SYS ED,1,0:SLEEP1        MG
·220 SYS ED,0,1:PRINTEQ$"[RIGHT]TO WINDOW
[3"."]":SLEEP1                                DA
·230 SYS ED,2,1:LIST:SLEEP2                   DG
·240 SYS ED,0,1:PRINTEQ$"[RIGHT]PRINT ANY
WHERE[3"."]":SLEEP1                           PB
·250 SYS ED,1,1:PRINT"[DOWN][RVSON]HERE,"
;:SLEEP1:SYS AT,0,12,0,,EQ$"HERE,";           FC
·260 SLEEP1:PRINT" OR HERE,":SLEEP1           NO
·270 SYS ED,0,1:PRINTEQ$"[RIGHT]AND RETUR
N."                                           NM
·280 SLEEP3                                   EO
·290 PRINT"[RVSON]";:COLOR5,2                 PB
·300 WINDOW4,2,24,20,1:SYS FR,,,,"[3"[RI
GHT]"]MENU"                                   OF
·310 PRINT:PRINT"[RIGHT]1. RESTART":PRINT
"[RIGHT]2. THE SHADOW"                        DJ
```

```
·320 PRINT"[RIGHT]3. THE TUNNEL":PRINT"[R
IGHT]4. QUIT"                                 JG
·330 PRINT:PRINT:PRINT"[RIGHT]YOUR CHOICE
?";                                           GD
·340 SYS 52591:REM CRSR ON                    KP
·350 GETKEY K$:IF K$<"1" ORK$>"4"THENPRIN
T"[3"[UP]"]":GOTO330                          MH
·360 SYS 52639,32:REM CRSR OFF                IL
·370 PRINTK$                                  HK
·380 K=VAL(K$):ONK GOTO110,410,480            LJ
·390 SLOW:END                                 GO
·400 REM SIMULATE A "FLOATING" WINDOW WIT
H SHADOW                                      EM
·410 WINDOW7,8,SW-9,14:PRINT"[BLACK][RVSO
N]";:AL=PEEK(FA):AH=PEEK(FA+1)                IP
·420 POKEFA,0:POKEFA+1,11:REM POINT TO SP
ACE CHARACTERS                                KG
·430 SYS FR:WINDOW8,7,SW-8,13                 BN
·440 COLOR5,8:PRINT"[RVSON]";:SYS FR          BF
·450 POKE FA,AL:POKE FA+1,AH:REM DEFAULT
FRAME                                         KH
·460 SLEEP3:GOTO280                           GJ
·470 REM TUNNEL                               DJ
·480 COLOR5,14                                IK
·490 FORI=0TO7:SYS FR:NEXT                    IP
·500 GOTO280                                  BN
·510 BG=0:SW=RWINDOW(2)                       EC
·520 SA=4864:FR=SA:AT=SA+3:CR=SA+6:ED=SA+
9:FA=SA+172                                   OP
·530 M$="[RIGHT][RVSON]WINDOW":B1$="[9" "
]"                                            EN
·540 FORI=0TO8:POKE2816+I,ASC(MID$(B1$,I+
1,1))                                         KH
·550 NEXT                                     KK
·560 IFSW=80THENBG=6:FAST                     OG
·570 EQ$="[UP]"+CHR$(27)+"Q"                  BN
·580 RETURN                                   NE
```

GUERILLA

FROM PAGE 16

```
·10 POKE53281,.:POKE53280,.:PRINTCHR$(8)C
HR$(142)"[CLEAR][7"[DOWN]"]"TAB(15)"[PUR
PLE]GUERILLA"                                 DE
·20 PRINTTAB(8)"[CYAN][DOWN][DOWN](C)1988
 CLEVE BLAKEMORE"                             NK
·30 DIM T(7),B(7),E$(4),D$(103),I$(7)         HA
·40 V=53248:SP=2040:J=56320:S=54272           BJ
·45 M=90:I=1:N=16:C=.:G=30:U=1:L=7:P=5318
0:AM=.:BM=4:TH=.:LM=50                        MH
·50 K=21:B=255:W=13:Q=V+39:R=2:E=5:XC=37:
YC=19:H=3:XS=7:YS=24:Z=150:LV=1:DM=103        BH
·60 FORX=2TO15:READA:POKEV+X,A:NEXT:FORX=
.TO127:READA:POKE832+X,A:NEXT                 NB
·70 FORX=.TO63:POKE11*64+X,.:POKE15*64+X,
.:NEXT                                        GB
·80 FORX=.TO12STEP3:POKE11*64+29+X,24:NEX
T                                             BC
·90 POKE11*64+35,255:POKESP,11:POKEV,156:
POKEV+1,188:POKEV+39,1                        FC
·100 FORX=53000TO53223:READA:POKEX,A:NEXT     BB
·110 FORX=UTOL:B(X)=2[UPARROW]X:NEXT          PH
·120 A$="[RVSON][c Y][s P][RVSOFF][sEP][U
P][3"[LEFT]"][RVSON][sEP][s C][s N][DOWN
]":E$(.)=" [UP] [DOWN][DOWN][LEFT] ":E$(
1)=" [DOWN]  ":E$(2)=" [DOWN]  "              MJ
·130 E$(3)=" [UP][LEFT]  ":E$(4)=" [UP] [
DOWN] ":BL$="[RVSON][RED]"                    PG
·140 B$="[c 4][RVSON]":FORX=.TO19:B$=B$+"
[s O][s P]":BL$=BL$+CHR$(32):NEXT:C$=RIG
HT$(B$,39)+"[s O]"                            JA
·150 I$(.)="[RVSOFF] ":I$(1)="[RVSOFF][c
G]":I$(2)="[RVSOFF][c H]":I$(3)="[RVSOFF
][c J]":I$(4)="[RVSOFF][c K]":I$(5)="[RV
SON][c L]"                                    NM
·160 I$(5)="[RVSON][c N]":I$(6)="[RVSON][
c M]":I$(7)="[RVSON] ":V$="ENEMY DECIMAT
ED[3"."]FORTRESS SAVED!!"                     EJ
·170 FORX=.TO39:D$=D$+"[s I]":E$=E$+"[s K
]":NEXT                                       BF
·180 FORX=.TO16:FORA=.TO7:D$(X*6+A)=LEFT$
(BL$,X+2)+I$(A)+"[RVSOFF]":NEXT:NEXT          GD
·190 POKEV+16,64+128:FORX=1TO7:POKESP+X,1
3:POKEV+39+X,5:NEXT                           JF
·200 GOSUB480:GOSUB450:GOSUB460:X=FRE(.)      GD
·210 FORX=.TO103:SYSP,9,23:PRINTD$(X):NEX
T                                             IJ
·220 X=1:POKEV+K,U:C=PEEK(V+G):C=PEEK(V+G
):SYS53000                                    ND
·230 REM MAIN LOOP                            OE
·240 F=PEEK(J)ANDN:X=X+U+(X=L)*L:IFT(X)TH
ENT(X)=T(X)-U:IFT(X)=.THENGOSUB290            HO
·250 C=RND(U)*M+U:IFC<L+UTHENIFT(C)=.THEN
T(C)=G-INT(I):POKEV+K,PEEK(V+K)ORB(C)         IA
·260 C=PEEK(V+G):ON-(F>.)GOTO240:GOSUB390
:AM=AM+U                                      BD
·265 IFAM=LMTHENAM=.:BM=BM-U:SYSP,27+H*BM
,YS:PRINT"[BLACK]"A$;:ON-(BM=.)GOTO320        EO
·270 ON-(C=.)GOTO240:FORA=UTOL:IF(CANDB(A
))THENGOSUB340                                JI
·280 NEXT:C=PEEK(V+G):C=PEEK(V+G):GOTO240     LB
·290 SYSP,RND(U)*XC,RND(U)*H+YC:PRINTE$(R
ND(U)*E);:POKEQ+X,W:GOSUB400:GOSUB390         BB
·300 DM=DM-U:SYSP,9,23:PRINTD$(DM)" ":FOR
A=.TON:POKEV+G+H,A:NEXT:A=X:IFDMGOTO380       PD
·310 POKES+18,129:FORX=.TO400:POKES+15,RN
D(.)*Z:POKEV+33,XANDG:NEXT:GOSUB450           ID
·320 POKEV+21,.:PRINT"[CLEAR][5"[DOWN]"]"
:PRINTTAB(4)"[c 3]OUR FORTRESS HAS BEEN
DESTROYED !!"                                 LF
·330 PRINTTAB(12)"[c 5][3"[DOWN]"]FINAL S
```

```
CORE:[WHITE]"SC:FORX=.TO7000:NEXT:RUN    IE
·340 POKESP+A,W+U:POKEQ+A,R:SC=SC+Z*LV:SY
SP,XS,YS:PRINT"[WHITE][RVSOFF]"SC;:M=M+(
M>L)                                     DJ
·350 IFI<26THENI=I+.5                    EN
·360 IFI=100RI=180RI=25THENLV=LV+U:POKE20
05,48+LV:GOSUB420                        KM
·370 TH=TH+U:IFTH=ZTHENPOKEV+21,.:PRINT"[
CLEAR][6"[DOWN]"]"TAB(11)V$:GOSUB420:GOT
0330                                     KF
·380 GOSUB410:POKEV+K,PEEK(V+K)AND(B-B(A)
):POKESP+A,W:POKEQ+A,E:T(A)=.:RETURN     MC
·390 POKES+8,4:POKES+11,129:FORA=.TO50:NE
XT:POKES+11,128:RETURN                   EF
·400 POKES+18,129:FORA=ZTO.STEP-E:POKES+1
5,A:NEXT:POKES+18,128:RETURN             PM
·410 FORD=.TOZ:NEXT:RETURN               MM
·420 M$="2513314215024215O2":GOSUB450:POK
ES+6,96:FORLL=1TO18STEP3                 MC
·430 POKES+1,VAL(MID$(M$,LL,2)):POKES+4,1
7:FORD=1TO90*VAL(MID$(M$,LL+2,1)):NEXT   DH
·440 POKES+4,16:NEXT:GOSUB450:GOSUB460:RE
TURN                                     PN
·450 FORLL=STOS+23:POKELL,.:NEXT:POKES+24
,15:RETURN                               GK
·460 POKES+6,73:POKES+13,234:POKES+20,234
:RETURN                                  KH
·470 REM GAME SCREEN                     DB
·480 PRINT"[CLEAR][5"[RIGHT]"][c 8][RVSON
][5" "][WHITE][c *]"                     CF
·490 PRINT"[5"[RIGHT]"][c 8][RVSON] [RVSO
FF][3" "][RVSON] [WHITE] [RVSOFF]  [GREE
N][cEP] [cEP] [3"[cEP]"] [c 5][RVSON] [c
 4][c *][RVSOFF][3" "][c 1][RVSON][s P][
s O][s P][s O][s P][s O][s P][s O][s P][
s O][s P][c 2][c *]"                     PI
·500 PRINT"[5"[RIGHT]"][c 8][RVSON] [RVSO
FF][3" "][RVSON] [WHITE] [RVSOFF] [GREEN
][cEP][cEP][c +][cEP][c +][cEP][c +][cEP
][c +][c 5][RVSON] [c 4] [c *][RVSOFF]
[c 1][RVSON][s O][s P][s O][s P][s O][s
P][s O][s P][s O][s P][s O][c 2] [c *]"  LB
·510 PRINT"[5"[RIGHT]"][c 8][RVSON] [RVSO
FF][3" "][RVSON] [WHITE] [RVSOFF]  [GREE
N][cEP][c -][c +][cEP][RVSON][c +][RVSOF
F][c +][RVSON][c +][RVSOFF][cEP][c 5][RV
SON] [c 4]  [c 8][3" "][c 1][s O][s P][s
 O][s P][RVSOFF][3" "][RVSON][s P][s O][
s P][c 2]  "                             AI
·520 PRINT"[RIGHT][RIGHT][c 5][RVSON] [9"
[c S]"] [c 8][c *][c 2][RVSOFF][s M][s V
][s V][s N][s M][s V][s V][c 5][RVSON] [
c 4]  [c 8]  [c 1][s O][s P][s O][s P][s
 O][RVSOFF][3" "][RVSON][s O][s P][s P][
c 2]  "                                  HL
·530 PRINT"[RIGHT][RIGHT][c 5][RVSON] [6"
[c S]"][RVSOFF][3" "][RVSON] [c 8] [c *]
[c 2][c I] [c *][sEP] [RVSOFF][s O][c 5]
[RVSON] [c 4]  [c 8]  [c 1][s P][s O][s
P][s O][s P][RVSOFF][3" "][RVSON][s P][s
 O][s P][c 2]  "                         KC
·540 PRINT"[RIGHT][RIGHT][c 5][RVSON] [6"
[c S]"][RVSOFF][3" "][RVSON] [c 8]  [RVS
OFF] [c 2][c *][RVSON]  [RVSOFF][sEP] [c
 1][RVSON][s P][c 4]  [c 8] [c 1][s P][s
 O][s P][s O][s P][s O][s P][s O][s P][s
 O][s P][s O][c 2]  "                    FA
·550 PRINT"[RIGHT][RIGHT][c 5][RVSON] [6"
[c S]"][RVSOFF][3" "][RVSON] [c 8]  [RVS
OFF]  [c 2][RVSON][c H][c L][RVSOFF]  [c
 1][RVSON][s O][c 4] [c 1][s O][s P][s O
][s P][s O][s P][s O][s P][s O][s P][s O
][s P][s O][s P][c 2]  "                 CC
·560 PRINT"[RIGHT][RIGHT][c 5][RVSON] [9"
[c S]"] [c 8]  [RVSOFF]  [c 2][RVSON][c
G][c N][RVSOFF]  [c 1][RVSON][s P][s O][
s P][s O][s P][s O][s P][s O][s P][s O][
s P][s O][s P][s O][s P][s O][c 2]  "    BB
·570 PRINT"[RIGHT][GREEN][c -][c 5][RVSON
] [RVSOFF][3" "][RVSON][6"[c S]"] [c 8]
 [GREEN][RVSOFF][c -] [c 2][RVSON][s Q][
c L][RVSOFF]  [c 1][RVSON][s O][s P][s O
][RVSOFF][3" "][RVSON][s O][s P][s O][s
P][RVSOFF][3" "][RVSON][s P][s O][s P][c
 2]  "                                   BD
·580 PRINT"[GREEN][c -][c -][c 5][RVSON]
[RVSOFF][3" "][RVSON][6"[c S]"] [c 8]  [
GREEN][RVSOFF][c -][c -][c 2][RVSON][c H
][c M][RVSOFF]  [c 1][RVSON][s P][s O][s
 P][RVSOFF][3" "][RVSON][s P][s O][s P][
s O][RVSOFF][3" "][RVSON][s O][s P][s O]
[c 2]  "                                 OO
·590 PRINT"[GREEN][c -][RVSON][c +][c 5]
[RVSOFF][3" "][RVSON][6"[c S]"] [c 8]  [
GREEN][c +][RVSOFF][cEP][c 2][RVSON]  [R
VSOFF]  [c 1][RVSON][s O][s P][s O][RVSO
FF][3" "][RVSON][s O][s P][s O][s P][RVS
OFF][3" "][RVSON][s P][s O][s P][c 2]  " KN
·600 PRINT"[GREEN][c +][c +][c 5][RVSON]
[9"[c S]"] [c 8]  [GREEN][c +][RVSOFF][c
 +][c 2][RVSON] [c M][RVSOFF] [GREEN][c
-][c 1][RVSON][s P][s O][s P][s O][s P][
s O][s P][s O][s P][s O][s P][s O][s P][
s O][s P][s O][c 2]  "                   AD
·610 PRINT"[GREEN][RVSON][c +][RVSOFF][c
 +][c 5][RVSON] [6"[c S]"][RVSOFF][3" "][
RVSON] [c 8]  [GREEN][RVSOFF][c +][RVSON
][c +][RVSOFF][c +][RVSON][c +][RVSOFF][
c -][RVSON][c +][c 1][s O][s P][s O][s P
][s O][s P][s O][s P][s O][s P][s O][s P
][s O][s P][s O][s P][c 2]  "            HN
·620 PRINT"[GREEN][c +][RVSON][c +][c 5]
[6"[c S]"][RVSOFF][3" "][RVSON] [c 8]  [
GREEN][4"[c +]"][RVSOFF][c +][RVSON][c +
][c 1][s P][s O][s P][s O][s P][s O][s P
][s O][s P][s O][s P][s O][s P][s O][s P
```

```
][s O][c 2]  "                                     CO
·630 PRINT"[GREEN][c +][c +][c 5][RVSON]
[6"[c S]"][RVSOFF][3" "][RVSON] [c 8]  [
GREEN][c +][RVSOFF][c +][RVSON][4"[c +]"
][c 1][s O][s P][s O][s P][s O][s P][s O
][s P][s O][s P][s O][s P][s O][s P][s O
][s P][c 2]  "                             GM
·640 PRINT"[c 5]"D$E$B$C$B$C$B$;"[RVSOFF]
[WHITE]GUERILLA"                           EC
·650 PRINT"[c 8]SCORE [WHITE]=[7" "][c 7]
ATTACK [WHITE]1[RIGHT][c 5]AMMO[c 2]"A$A
$A$A$;"[HOME]":RETURN                      MK
·660 REM SPRITE LOCATION DATA              HG
·670 DATA96,93,72,61,48,125,96,158,217,12
6,17,126,17,78:REM V+16=64+128             LJ
·680 REM SPRITE DATA                       CD
·690 DATA1,252,,3,254,,7,199,,7,56,128,6,
221,128,1,125,.                            IG
·700 DATA6,21,,4,46,128,4,29,96,12,35,104
,28,62,220,60,62,62                        AG
·710 DATA124,56,31,124,54,15,121,45,15,11
4,155,239,4,87,239,10,175,222              HA
·720 DATA19,215,156,32,107,124,193,245,12
0,,,,126,18,15,195,9,23                    EM
·730 DATA97,36,63,225,18,63,225,8,51,222,
1,179,128,7,222,180,15,225                 HK
·740 DATA110,31,126,223,62,252,191,61,248
,95,1,240,95,126,104,95,185,216            FH
·750 DATA223,85,183,190,147,110,136,16,22
1,240,3,187,240,4,117,80,15,250,80,.       JE
·760 REM INTERRUPT ROUTINE                 IG
·770 DATA 120,169,31,141,20,3,169,207,141
,21,3,169,166,141,17                       IN
·780 DATA 3,169,207,141,18,3,88,96,173,,2
20,41,15,133,253                           LB
·790 DATA 169,15,56,229,253,168,185,221,2
07,240,8,141,168,2,160                     AL
·800 DATA 0,32,60,207,76,163,207,173,168,
2,10,170,189,202,207                       NJ
·810 DATA 8,152,10,170,40,48,25,240,43,17
3,220,207,24,125,0                         KN
·820 DATA 208,157,,208,144,31,169,1,77,16
,208,141,16,208,76                         MI
·830 DATA 120,207,189,,208,56,237,220,207
,157,,208,176,8,169                        AG
·840 DATA 1,77,16,208,141,16,208,173,168,
2,10,170,232,189,202                       II
·850 DATA 207,8,152,10,170,232,40,48,15,2
40,23,173,220,207,24                       FF
·860 DATA 125,,208,157,,208,76,162,207,18
9,,208,56,237,220                          MF
·870 DATA 207,157,,208,96,76,49,234,173,,
220,41,15,133,253                          CL
·880 DATA 169,15,56,229,253,168,185,221,2
07,168,169,,108,5,0                        IN
·890 DATA 32,155,183,138,72,32,155,183,10
4,168,24,76,240,255,0                      MC
·900 DATA 0,,255,1,255,1,,1,1,,1,255,1,25
5,0                                        MB
·910 DATA 255,255,3,,1,5,,7,8,6,,3,2,4     KF
```

TOMB OF HORROR

FROM PAGE 48

BASIC PORTION

```
·10 IFX=.THENX=1:LOAD"TOMB.OBJ",8,1        HH
·20 POKE56,144:POKE55,0:CLR                BD
·30 POKE53280,.:POKE53281,.:X=RND(-TI):FO
RX=901TO970:READA:POKEX,A:NEXT             MJ
·40 FORX=828TO839:READA:POKEX,A:NEXT:FORX
=679TO753:READA:POKEX,A:NEXT:GOTO790       AG
·50 D=RND(U)*G:POKES+K,U(D):POKES+7,V(D):
POKES+G,U(D+K+F):POKES+G-U,V(D+K+F)        DM
·60 RETURN                                 MK
·70 VL=FC:ONFC+UGOSUB450,450,460,470,480   DN
·80 RETURN                                 NO
·90 ONFC+UGOSUB500,510,520,530:RETURN      LJ
·100 ONFC+UGOSUB540,550,560:RETURN         CL
·110 PRINT"[HOME][BLACK][RVSON] +[36" "]/
 ";                                        FG
·120 PRINT"  [RVSOFF][36"[SS]"][RVSON]  "
;                                          AI
·130 PRINT"  [RVSOFF][36"[SS]"][RVSON]  "
;                                          DC
·140 PRINT"  [RVSOFF][36"[SS]"][RVSON]  "
;                                          DM
·150 PRINT"  [RVSOFF][36"[SS]"][RVSON]  "
;                                          FG
·160 PRINT"  [RVSOFF][4"[SS]"]"Z$"[RVSON]
([BLACK][26" "]"Z$")[BLACK][RVSOFF][4"[S
S]"][RVSON]  ";                            LB
·170 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
"Z$"([BLACK][24"="]"Z$")[BLACK][SS][RVSO
FF][4"[SS]"][RVSON]  ";                    JF
·180 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS] +[20" "]/ [SS][SS][RVSOFF][4"[SS]"]
[RVSON]  ";                                BJ
·190 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][20"[SS]"][RVSON]  [SS][SS
][RVSOFF][4"[SS]"][RVSON]  ";              LO
·200 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS]"Z$"[RVSON]([BLACK
][14"="]"Z$")[BLACK][RVSOFF][SS][SS][RVS
ON]  [SS][SS][RVSOFF][4"[SS]"][RVSON]  "
;                                          PP
·210 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS][RVSON][SS]+[12" "
]/[SS][RVSOFF][SS][SS][RVSON]  [SS][SS][
RVSOFF][4"[SS]"][RVSON]  ";                PH
·220 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS][RVSON][SS] [RVSOF
F][12"[SS]"][RVSON] [SS][RVSOFF][SS][SS]
```

```
[RVSON]  [SS][SS][RVSOFF][4"[SS]"][RVSON
]  ";                                    FI
•230 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS][RVSON][SS] [RVSOF
F][SS]"Z$"[RVSON]([BLACK][8"="]"Z$")[BLA
CK][RVSOFF][SS][RVSON] [SS][RVSOFF][SS][
SS][RVSON]  [SS][SS][RVSOFF][4"[SS]"][RV
SON]  ";                                 HF
•240 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS][RVSON][SS] [RVSOF
F][SS][RVSON][SS] "Z$"[RVSOFF][c +][4"[c
 Z]"][s *][BLACK][RVSON] [SS][RVSOFF][SS
][RVSON] [SS][RVSOFF][SS][SS][RVSON]  [S
S][SS][RVSOFF][4"[SS]"][RVSON]  ";       NP
•250 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS][RVSON][SS] [RVSOF
F][SS][RVSON][SS] "Z$"[RVSOFF][PI][BLACK
][4" "]"Z$"[RVSOFF][c *][BLACK][RVSON] [
SS][RVSOFF][SS][RVSON] [SS][RVSOFF][SS][
SS][RVSON]  [SS][SS][RVSOFF][4"[SS]"][RV
SON]  ";                                 FE
•260 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS][RVSON][SS] [RVSOF
F][SS][RVSON][SS] "Z$"[RVSOFF][PI][BLACK
][4" "]"Z$"[RVSOFF][c *][BLACK][RVSON] [
SS][RVSOFF][SS][RVSON] [SS][RVSOFF][SS][
SS][RVSON]  [SS][SS][RVSOFF][4"[SS]"][RV
SON]  ";                                 EP
•270 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS][RVSON][SS] [RVSOF
F][SS]"Z$"[RVSON]<[BLACK][8" "]"Z$">[BLA
CK][RVSOFF][SS][RVSON] [SS][RVSOFF][SS][
SS][RVSON]  [SS][SS][RVSOFF][4"[SS]"][RV
SON]  ";                                 HB
•280 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS][RVSON][SS]/[12" "
]+[SS][RVSOFF][SS][SS][RVSON]  [SS][SS][
RVSOFF][4"[SS]"][RVSON]  ";              JI
•290 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]  [RVSOFF][SS][SS]"Z$"[RVSON]<[BLACK
][14" "]"Z$">[BLACK][RVSOFF][SS][SS][RVS
ON]  [SS][SS][RVSOFF][4"[SS]"][RVSON]  "
;                                        BJ
•300 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS] /[20" "]+ [SS][SS][RVSOFF][4"[SS]"]
[RVSON]  ";                              IM
•310 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
[SS]/[22" "]+[SS][SS][RVSOFF][4"[SS]"][R
VSON]  ";                                EK
•320 PRINT"  [RVSOFF][4"[SS]"][RVSON][SS]
"Z$"<[BLACK][24" "]"Z$">[BLACK][SS][RVSO
FF][4"[SS]"][RVSON]  ";                  BP
•330 PRINT"  [RVSOFF][4"[SS]"]"Z$"[RVSON]
<[BLACK][26" "]"Z$">[BLACK][RVSOFF][4"[S
S]"][RVSON]  ";                          MB
•340 PRINT" /[36" "]+ ";                 BI
•350 PRINTD$B$D$I$(DL(CL,T))TAB(G+H)"[WHI
TE]"D$(CD*-(CU=G))"[c 3]";               LG
•360 PRINTTAB(39-LEN(T$(DL(CL,U))))T$(DL(
CL,U));:SYSRL:RETURN                     EN
•370 PRINT"[HOME][RVSOFF][DOWN]  ":FORX=.
TO20:PRINT"  ":NEXT:RETURN               GC
•380 PRINT"[HOME][RVSOFF][7"[DOWN]"]":FOR
X=.TO10:PRINTTAB(8)"  ":NEXT:RETURN      FN
•390 PRINT"[HOME][RVSOFF][11"[DOWN]"]";TA
B(13)" [DOWN][LEFT]";:FORX=.TOF:PRINT" [
DOWN][LEFT]";:NEXT:RETURN                EH
•400 PRINT"[HOME][13"[DOWN]"]";TAB(16)" [
DOWN][LEFT] [DOWN][LEFT] ":RETURN        CE
•410 PRINT"[HOME][RVSOFF]":FORX=.TO21:PRI
NTTAB(39)CHR$(20)CHR$(32):NEXT:RETURN    PN
•420 PRINT"[HOME][RVSOFF][7"[DOWN]"]":FOR
X=.TO10:PRINTTAB(30)"  ":NEXT:RETURN     AA
•430 PRINT"[HOME][10"[DOWN]"]":FORX=.TO5:
PRINTTAB(26)" ":NEXT:RETURN              BO
•440 PRINT"[HOME][RVSOFF][13"[DOWN]"]";TA
B(23)" [DOWN][LEFT] [DOWN][LEFT] ":RETUR
N                                        MM
•450 PRINT"[HOME][5"[DOWN]"][RVSOFF]";:FO
RX=.TO17:PRINTTAB(6)W1$:NEXT:RETURN      IO
•460 PRINT"[HOME][9"[DOWN]"][RVSOFF]";:FO
RX=.TO9:PRINTTAB(12)W2$:NEXT:RETURN      BP
•470 PRINT"[HOME][11"[DOWN]"][RVSOFF]";:F
ORX=.TO5:PRINTTAB(15)W3$:NEXT:RETURN     PI
•480 PRINT"[HOME][13"[DOWN]"][RVSOFF]";:F
ORX=.TOT:PRINTTAB(17)W4$:NEXT:RETURN     ED
•490 RETURN                              HJ
•500 PRINT"[HOME][23"[DOWN]"]"TAB(13)S1$;
:RETURN                                  MP
•510 PRINT"[HOME][20"[DOWN]"]"TAB(15)S2$;
:RETURN                                  IB
•520 PRINT"[HOME][17"[DOWN]"]"TAB(17)S3$;
:RETURN                                  JD
•530 PRINT"[HOME][16"[DOWN]"]"TAB(18)S4$;
:RETURN                                  IH
•540 PRINT"[HOME]"TAB(13)E1$:RETURN      MG
•550 PRINT"[HOME][7"[DOWN]"]"TAB(15)E2$:R
ETURN                                    ON
•560 PRINT"[HOME][10"[DOWN]"]"TAB(17)E3$:
RETURN                                   FB
•570 IF(DL(CL,.)ANDB(F+H))=.THEN610      MF
•580 SB=PEEK(V+21):POKEV+21,.:GOSUB2730:G
OSUB2510:DL(CL,.)=(DL(CL,.)ANDG)         GH
•590 X=INT(RND(U)*10*(DL+U)):C$(1)="YOU T
AKE"+STR$(X)+" HP DAMAGE!":AA=18:N=1     EI
•600 LF=LF-X:GOSUB2570:GOSUB3120:FORX=.TO
2000:NEXT:GOSUB2880:POKEV+21,SB          OA
•610 PRINTD$TAB(G+H)"MOVE!";:CEM=FNTE(.):
POKESE,CEM:GOSUB110:PRINTZ$;             JA
•620 FC=F:VL=FC:LV=CD-U+F*-(CD=U):RV=CD+U
+F*(CD=F)                                JL
•630 EBM=FNCP(.):IFEBM>.ANDEBM<DETHEN650  JD
•640 FC=FC-U:ON-(FC>-U)GOTO630:RETURN    KK
•650 EL=FNCP(LV):ER=FNCP(RV):IFDL(EBM,.)T
HEN670                                   FB
•660 GOSUB70:FC=FC-U:ONU-(FC<.)GOTO630,71
```

```
0                                            CI
·670 IFDL(EL,.)ANDFC<FTHENONFC+UGOSUB370,
380,390,400                                  LJ
·680 IFDL(ER,.)ANDFC<FTHENONFC+UGOSUB410,
420,430,440                                  AO
·690 FORD=.TOH:IF(DL(EBM,.)ANDB(D))THENON
DGOSUB90,100,70:PRINTZ$;                     HE
·700 PRINT"[RVSOFF]";:NEXT:FC=FC-U:IFFC>-
UTHEN630                                     LD
·710 GOSUB720:CSM=FNSW(.):POKEV+24,(PEEK(
V+24)AND15)ORCSM:POKEV+21,EL:RETURN          ID
·720 IFFCANDVLTHENVL=VL-U                    MO
·730 D=191:POKEFNMP(U),D:POKEFNMP(T),D:PO
KEFNMP(H),D:POKEFNMP(F+U),D                  HK
·740 EL=.:ER=.:POKEV+21,.:FORX=VLTO.STEP-
U:FC=FNCM(X):IFDL(FC,U)=.THEN780             JH
·750 FC=ABS(DL(FC,U)-U):EL=(ELORSP(X+F))     OJ
·760 MB=SP(X)-U:FORD=UTOLEN(SB$(X,FC)):PO
KEFNMP(MB+D),ASC(MID$(SB$(X,FC),D,U))        BC
·770 POKEBC+SP(X)+D,CS(FC):NEXT              HA
·780 NEXT:SB=EL:RETURN                       HN
·790 PRINTCHR$(147)CHR$(8)CHR$(142)"[BLUE
]STAND BY[4"."]":GOSUB2900:GOSUB2980         OG
·800 DIMDL(383,3),B(15),SP(7),SB$(3,12),D
$(4),CS(12),L$(6),I(30),M(24),MH(24)         OE
·810 DIMCM$(14),T$(13),I$(30),C$(30),C(30
),M$(44),LV(2,2),CZ$(2),H(8),SM$(8)          KK
·820 DIMTS(24),U(26),V(26)                   MN
·830 FC=.:CL=17:CD=2:VL=.:EL=.:ER=.:VS=.:
LF=20:M=.:W=.005:Z=.94:WL=17                 LP
·840 TL=388:TGL=16:CSM=.:CEM=192:DE=383:V
=53248:SB=.:RL=828:SE=648:SR=679             PM
·850 U=1:T=2:H=3:F=4:K=8:G=15:JOY=56320:D
L=.:DM=.:AR=.:CU=.:WE=.                      HH
·860 S=54272:BC=V+38:SS=.:SC=.:CC=55296:S
Z=959:I=.                                    LM
·870 D(1)=1:D(2)=16:D(3)=-1:D(4)=-16         DM
·880 H(.)=1:H(1)=10:H(2)=2:H(3)=2:H(4)=1:
H(5)=7:H(6)=3:H(7)=13:H(8)=5                 DJ
·890 SP(.)=.:SP(1)=4:SP(2)=6:SP(3)=7:SP(4
)=15:SP(5)=48:SP(6)=64:SP(7)=128             DM
·900 D$(.)="  ?  ":D$(1)="EAST ":D$(2)="S
OUTH":D$(3)="WEST ":D$(4)="NORTH"            AJ
·910 SM$(.)="[EP][EP]":FORX=1TO8:SM$(X)=C
HR$(32)+CHR$(32):NEXT:CR$=CHR$(13)           AG
·920 FORX=.TO38:W$=W$+CHR$(160):B$=B$+CHR
$(32):MB$=MB$+"*":NEXT                       ON
·930 D$="[HOME][c 8][RVSOFF]":FORX=.TO23:
D$=D$+"[DOWN]":NEXT                          LB
·940 W1$=LEFT$(W$,28):W2$=LEFT$(W$,16):W3
$=LEFT$(W$,10):W4$=LEFT$(W$,6)               GP
·950 S1$="[RVSON][BLACK])"+LEFT$(W$,12)+"
(":S2$="[RVSON][BLACK])"+LEFT$(W$,8)+"("    CC
·960 S3$="[RVSON][BLACK])"+LEFT$(W$,4)+"(
":S4$="[RVSON][BLACK])"+LEFT$(W$,2)+"("      FB
·970 E1$="[RVSON][BLACK]>"+LEFT$(W$,12)+"
<":E2$="[RVSON][BLACK]>"+LEFT$(W$,8)+"<"    IM
·980 E3$="[RVSON][BLACK]>"+LEFT$(W$,4)+"<
":CZ$(.)="[c 4]":CZ$(1)="[BLUE]":CZ$(2)=
"[c 2]"                                      HL
·990 I$(.)=LEFT$(B$,15):T$(.)=LEFT$(B$,13
)                                            CM
·1000 DEF FNCP(X)=FC*D(CD)+(CL+D(X)):DEF
FNSW(X)=(TGL-CSM)                            EP
·1010 DEF FNTE(X)=(TL-CEM):DEF FNCM(X)=X*
D(CD)+CL:DEF FNMP(X)=CEM*256+1016+X          OP
·1020 DEFFNJ(X)=(G-(XANDG)):DEFFNC(X)=(LF
<K)*-T+(LF>KANDLF<G)*-(F+H)                  JP
·1030 DEFFNCD(X)=CD+(CD=F+U)*F+(CD=.)*-F:
DEFFNJF(X)=((JYAND(G+U))=.)                  DH
·1040 DEFFNH(A)=(AR+I(AR)*T+LF/K)            OJ
·1050 FORX=.TO15:B(X)=2[UPARROW]X:NEXT       KA
·1060 FORX=.TODE:READD:DL(X,.)=D:NEXT:FOR
X=.TO15:READA:POKEV+X,A:NEXT                 IP
·1070 FORX=1TO13:READT$(X):NEXT:FORX=.TO2
4:READA:M(X)=A:NEXT                          IB
·1080 FORX=.TO12:READCS(X):NEXTX:FORX=.TO
24:READTS(X):NEXT                            CC
·1090 FORX=.TO6:FORA=.TO3:SB$(A,X)=CHR$(3
2+X):NEXTA,X                                 EM
·1100 FORX=7TO12:FORA=.TO3:READSB$(A,X):N
EXTA,X                                       JL
·1110 FORX=.TO1:SB$(X,.)=CHR$(191)+CHR$(3
2):NEXT                                      BG
·1120 FORX=1TO30:READI$(X):NEXT:FORX=1TO1
4:READCM$(X):NEXT                            IP
·1130 FORX=.TO2:FORA=.TO2:READLV(X,A):NEX
TA,X:FORX=1TO44:READM$(X):NEXT               OD
·1140 J=1:FORD=.TOT:FORX=.TOK-U              GN
·1150 A=INT(RND(U)*96+(D*128+16)):ON-(DL(
A,.)=.)GOTO1150:DL(A,U)=M(J):M(J)=A          CB
·1160 MH(J)=J*F*(D+U):J=J+U:NEXT:NEXT:FOR
X=.TO63:POKE61376+X,.:NEXT                   FD
·1170 FORD=.TOT:FORX=.TO10                   FD
·1180 A=INT(RND(U)*96+(D*128+16)):ON-(DL(
A,.)=.)GOTO1180:DL(A,.)=DL(A,.)ORB(7)        DF
·1190 NEXT:NEXT                              ON
·1200 FORX=.TO26:A=2703*(2[UPARROW]((X-20
)/12)):IFA>65535THENA=65535                  LM
·1210 U(X)=INT(A/B(K)):V(X)=A-U(X)*B(K):N
EXT                                          IB
·1220 DL(WL,T)=U:DL(WL,H)=U:DL(WL+16,T)=1
0:DL(WL+16,H)=U                              MG
·1230 PRINTCHR$(147)CHR$(14):POKEV+21,.:P
OKEV+23,127:POKEV+29,15                      CB
·1240 POKEV+17,PEEK(V+17)OR64:GOSUB2890:G
OSUB610                                      JB
·1250 SS=144:SC=148:GOSUB2730                IG
·1260 C$(U)=M$(31):BO$="[c 4]":MC$="[WHIT
E]":N=U:GOSUB2540:GOSUB2880:GOSUB3050:IT
=IT+1000                                     CM
·1270 JY=PEEK(JOY):J=FNJ(JY):B=DL(CL,U):I
FB>.THENIFRND(U)+B/50>ZTHENGOSUB1590         PJ
·1280 ON-(RND(U)>Z)GOSUB50:ON-(FNJF(J)AND
```

```
AR>.)GOSUB1630                                    PN
·1290 IFJTHENONJGOSUB1380,1460,80,1490,80
,80,80,1500                                       DK
·1300 POKEV+K*F,FNC(.):LF=LF+W:ON-(TI>IT)
GOSUB1510:ON-(LF>.)GOTO1270                       NO
·1310 J=.:D=.:FORX=UTO100:POKE63488+32*K+
RND(U)*K,RND(U)*255+U:NEXT                        FP
·1320 PRINT"[BLACK][CLEAR]":POKEV+21,.:GO
SUB3130:FORX=250TO150STEP-U:SYSSR,.,240,
X,.,17,1                                          PN
·1330 NEXT:FORX=.TO7:POKE63488+32*K+X,.:N
EXT:X=.                                           AB
·1340 POKEV+17,PEEK(V+17)AND191                   BL
·1350 C$(1)=M$(43+X):C$(2)="":C$(3)="[WHI
TE][5" "]P L A Y[3" "]A G A I N ?[3" "]"          FG
·1360 BO$="[RED]":MC$="[c 3]":N=3:GOSUB25
60:GOSUB2520:IFI=UTHENPOKES+24,.:RUN30            GP
·1370 POKESE,4:POKE679,.:SYS679                   DL
·1380 IFDL(CL+D(CD),.)=.THENGOSUB3120:RET
URN                                               NA
·1390 CL=CL+D(CD):GOSUB570:RETURN                 PA
·1400 ON-(B>.)GOTO80:SB=PEEK(V+21):SS=144
:SC=148:GOSUB2730                                 IE
·1410 GOSUB2710:POKEV+21,.:GOSUB2740:SS=1
52:SC=156:GOSUB2730                               OK
·1420 GOSUB2710:GOSUB2790:ON-(I=G-U)GOTO1
450                                               PL
·1430 ONIGOSUB1800,1850,1930,1940,1950,19
60,2020,2100,2240,2320,2350,2430,2480             FL
·1440 SS=152:SC=156:GOSUB2880:GOTO1420            HO
·1450 SS=144:SC=148:GOSUB2880:POKEV+21,SB
:X=FRE(.):GOSUB50:RETURN                          HC
·1460 IFFNJF(.)THEN1400                           FN
·1470 IFDL(CL-D(CD),.)=.THENGOSUB3120:RET
URN                                               OH
·1480 CL=CL-D(CD):GOSUB570:RETURN                 KB
·1490 CD=CD-U:CD=FNCD(.):GOSUB570:RETURN          EL
·1500 CD=CD+U:CD=FNCD(.):GOSUB570:RETURN          EI
·1510 D=DL*K+U:J=D+K-U:FORX=DTOJ:ON-(MH(X
)=.)GOTO1580:A=-U                                 DM
·1520 IFCL<M(X)-KTHENA=M(X)+D(F)                  HP
·1530 IFCL>M(X)+KTHENA=M(X)+D(T)                  OO
·1540 IFCL>M(X)ANDCL<M(X)+KTHENA=M(X)+D(U
)                                                 IJ
·1550 IFCL<M(X)ANDCL>M(X)-KTHENA=M(X)+D(H
)                                                 JP
·1560 IFA<.ORA>DETHEN1580                         FJ
·1570 IFDL(A,.)ANDDL(A,U)=.THENDL(A,U)=AB
S(DL(M(X),U)):DL(M(X),U)=.:M(X)=A                 JA
·1580 NEXT:ON-(DL(CL,U)>.)GOSUB3030:GOSUB
730:POKEV+21,EL:IT=TI+SZ:RETURN                   AD
·1590 IFCU=20THENRETURN                           LP
·1600 IFWETHENB=B*((G-WE)/10)                     CP
·1610 LF=LF-B*T:GOSUB3120:FORX=.TOH:WAITV
+17,128:POKEV+34,H(X):POKEV+35,H(X)               OD
·1620 NEXT:GOSUB2890:POKEV+K*F,FNC(.):RET
URN                                               JN
·1630 ON-(B=.ORCU=20)GOTO80:E=CL:DM=FNH(.
):X=.                                             DD
·1640 ON-(MH(X)=.)GOTO1790:ON-(E<>M(X))GO
TO1790:MH(X)=MH(X)-DM:GOSUB3120                   AO
·1650 FORD=FTOK:WAITV+17,128:POKEV+34,H(D
):POKEV+35,H(D):NEXT:GOSUB2890                    IE
·1660 ON-(MH(X)>.)GOTO1790:MH(X)=.:DL(E,U
)=.:M(.)=.                                        AF
·1670 LV=.:A=-U                                   DH
·1680 A=A+U:FC=E+D(A)*LV:IFFC<.ORFC>DETHE
N1710                                             MG
·1690 IFDL(FC,.)=.ORDL(FC,T)THEN1710              MN
·1700 DL(FC,T)=TS(X):DL(FC,H)=U+INT(-(TS(
X)<G)*RND(U)*H):GOTO1730                          JB
·1710 IFA=FTHENA=.:LV=LV+U                        EA
·1720 GOTO1680                                    BG
·1730 FORX=.TOG:FORD=UTOF                         DM
·1740 POKEBC+D,X:NEXT:NEXT:POKEV+21,PEEK(
V+21)AND240                                       JF
·1750 GOSUB3090:LF=LF+B                           CJ
·1760 X=26+RND(U)*5:I(X)=I(X)+INT(RND(U)*
50*B*(LV+U))                                      JK
·1770 C$(1)="A "+M$(22+RND(U)*H)+" "+M$(2
5+RND(U)*H)+" OF "+I$(X)+"."                      PB
·1780 BO$="[YELLOW]":MC$="[WHITE]":N=U:GO
SUB2730:GOSUB2540:GOSUB2880:GOSUB350:RET
URN                                               OD
·1790 X=X+U:ON-(X<25)GOTO1640:RETURN              PE
·1800 IFDL(CL,T)=.THENC$(U)="NOTHING HERE
!":N=U:BO$="[PURPLE]":MC$="[WHITE]":GOSU
B2540:RETURN                                      EH
·1810 IFDL(CL,T)>14THEN1830                       NJ
·1820 IFI(DL(CL,T))THENC$(U)="YOU HAVE A
"+I$(DL(CL,T)):N=U:GOSUB2540:RETURN               NF
·1830 I(DL(CL,T))=I(DL(CL,T))+DL(CL,H):C$
(U)=I$(DL(CL,T))+" TAKEN!"                        AE
·1840 N=U:BO$="[WHITE]":DL(CL,T)=.:PRINTD
$I$(.);:GOSUB2540:RETURN                          CN
·1850 IFDL(CL,T)THENC$(U)="NOT ENOUGH ROO
M HERE.":N=U:BO$="[PURPLE]":MC$="[WHITE]
":GOTO2540                                        CB
·1860 GOSUB2580:IFX=-1THENRETURN                  IP
·1870 GOSUB2740:GOSUB2790:DL(CL,T)=C(I):D
L(CL,H)=I(C(I)):I(C(I))=.                         HA
·1880 C$(U)=I$(C(I))+" DROPPED.":N=U:BO$=
"[WHITE]":PRINTD$I$(C(I));                        IC
·1890 IFAR=C(I)THENAR=.                           EK
·1900 IFCU=C(I)THENCU=.                           FG
·1910 IFWE=C(I)THENWE=.                           DI
·1920 GOSUB2540:RETURN                            KE
·1930 GOSUB2630:ON-(N=.)GOTO2530:GOSUB274
0:GOSUB2790:AR=C(I):RETURN                        OH
·1940 GOSUB2670:ON-(N=.)GOTO2530:GOSUB274
0:GOSUB2790:CU=C(I):RETURN                        HC
·1950 GOSUB2650:ON-(N=.)GOTO2530:GOSUB274
0:GOSUB2790:WE=C(I):RETURN                        CO
·1960 IF(DL(CL,.)ANDF)=.THENC$(U)=M$(29):
```

```
BO$="[RED]":MC$="[YELLOW]":N=U:GOTO2540 MF
·1970 IFI(19)=.THENC$(U)=M$(30)+I$(19)+"!
":BO$="[PURPLE]":MC$="[CYAN]":N=U:GOTO25
40                                       EN
·1980 IFI(17)=.THENC$(U)=M$(30)+I$(17)+"!
":BO$="[PURPLE]":MC$="[c 4]":N=U:GOTO254
0                                        JD
·1990 ON-(CL<>WL)GOTO2010:IFI(25)=.THENC$
(U)=M$(42):BO$="[PURPLE]":MC$="[WHITE]":
N=U:GOTO2540                             NC
·2000 PRINTCHR$(147):GOSUB3090:X=U:GOTO13
40                                       AA
·2010 DL=DL-U:CL=CL-128:A=F:D=50:J=T:GOSU
B3070:GOTO2060                           KB
·2020 IF(DL(CL,.)ANDT)=.THENC$(U)=M$(28):
BO$="[RED]":MC$="[YELLOW]":N=U:GOTO2540  EO
·2030 IFI(19)=.THENC$(U)=M$(30)+I$(19)+"!
":BO$="[PURPLE]":MC$="[CYAN]":N=U:GOTO25
40                                       JB
·2040 IFI(17)=.THENC$(U)=M$(30)+I$(17)+"!
":BO$="[PURPLE]":MC$="[c 4]":N=U:GOTO254
0                                        GH
·2050 DL=DL+U:CL=CL+128:A=50:D=F:J=-T:GOS
UB3070                                   NO
·2060 SS=144:SC=148:GOSUB2880:Z$=CZ$(DL):
GOSUB610:GOSUB2890                       PN
·2070 GOSUB2710:GOSUB2730:GOSUB2740      FI
·2080 C$(U)=M$(31+DL):BO$="[c 4]":MC$="[W
HITE]":N=U:AA=U:SS=152:SC=156:GOSUB2730  JA
·2090 GOSUB2540:RETURN                   FN
·2100 PRINTD$B$D$"[YELLOW][RIGHT]"I$(AR)T
AB(17)"[RED]HP[YELLOW]"STR$(INT(LF))TAB(
39-LEN(I$(CU)))I$(CU);                   LH
·2110 IFWETHENC$(U)=I$(WE):BO$="[c 6]":MC
$="[GREEN]":N=U:AA=19:GOSUB2570          NC
·2120 GOSUB2580:IFX=-UTHENGOSUB350:RETURN BP
·2130 ONIGOTO2140,2140,2170,2170         CN
·2140 GOSUB2740:GOSUB2790:BO$="[WHITE]":C
$(U)="THE "+I$(C(I))+" IS OF"            EN
·2150 C$(T)=M$(18+I(C(I)))+" WORKMANSHIP.
":AA=K:N=T:LM=28:SM=20-LM/T              FO
·2160 GOSUB2740:GOSUB2550:GOSUB350:RETURN LC
·2170 GOSUB2740:GOSUB2790:IFC(I)=16THEN22
00                                       BF
·2180 C$(U)="YOU HAVE"+STR$(I(C(I)))+" "+
I$(C(I))+"."                             PC
·2190 N=U:BO$="[WHITE]":GOSUB2540:GOSUB35
0:RETURN                                 GF
·2200 C$(U)="UNROLLING PARCHMENT[3"."]":N
=U:BO$="[YELLOW]":MC$="[c 2]":GOSUB2560  NN
·2210 X=DL*128:J=U:FORA=.TO127STEP16:C$(J
)="":FORD=.TOG                           LP
·2220 C$(J)=C$(J)+SM$(DL(X+A+D,.)ANDG):NE
XT:C$(J+U)=C$(J):J=J+T                   FD
·2230 NEXT:N=J-U:BO$="[c 2]":MC$=CZ$(DL):
GOSUB2540:GOSUB350:RETURN                PG
·2240 IFCU=.THENC$(U)="YOU ARE EMPTY HAND
ED.":N=U:BO$="[c 8]":MC$="[BLUE]":GOTO25
40                                       OF
·2250 IFCU<21THENC$(U)="NOTHING HAPPENS."
:N=U:BO$="[c 4]":MC$="[c 8]":GOTO2540    MD
·2260 C$(U)=M$(15+(CU-21)):N=U:BO$="[YELL
OW]":MC$="[c 3]":GOSUB2560               GO
·2270 IFCU=21ANDDL(CL+D(CD),.)=KTHENDL(CL
+D(CD),.)=U                              HB
·2280 J=DL(CL+D(CD),U):IFCU=22AND(J=KORJ=
9ORJ=13)THEN2380                         FP
·2290 IFCU=23THENCU=.:I(23)=.:I(24)=U    IE
·2300 IFCU=24THENI(25)=U:I(24)=.:CU=.    ML
·2310 GOSUB2550:RETURN                   BP
·2320 IFI(18)=.THENC$(U)=M$(34):BO$="[c 7
]":MC$="[PURPLE]":N=U:GOSUB2540:RETURN   NG
·2330 I(18)=I(18)-U:LF=LF+10:C$(U)="YOUR
BODY IS FLOODED WITH STRENGTH":N=U       PF
·2340 BO$="[c 7]":MC$="[WHITE]":GOSUB2540
:RETURN                                  MC
·2350 J=ABS(DL(CL+D(CD),U)):IFJ=.THENC$(U
)=M$(35):MC$="[YELLOW]":N=U:GOSUB2540:RE
TURN                                     PF
·2360 IFJ<5ORJ=7THENC$(U)="THE IGNORANT B
RUTE IS SILENT.":N=U:BO$="[c 7]":GOTO254
0                                        LG
·2370 IFLF<B*100RJ>11ORRND(U)<.8THEN2400 KA
·2380 C$(U)=CHR$(34)+M$(14)+CHR$(34):C$(T
)="MURMURS THE "+T$(J)+"."               IJ
·2390 BO$="[PURPLE]":MC$="[YELLOW]":N=T:D
L(CL+D(CD),U)=-J:GOSUB2540:RETURN        DG
·2400 X=(RND(U)*F):C$(U)=CHR$(34)+"I'LL "
+M$(6+X)+"!!"+CHR$(34)                   CE
·2410 X=(RND(U)*F):C$(T)=M$(10+X)+" THE "
+T$(J)+"."                               AN
·2420 BO$="[RED]":MC$="[WHITE]":N=T:GOSUB
2540:RETURN                              IK
·2430 A=.:FORFC=.TOH:EBM=FNCP(.):IFEBM<.O
REBM>DETHEN2460                          LK
·2440 IF(DL(EBM,.)ANDB(7))=.THEN2460     DF
·2450 DL(EBM,.)=DL(EBM,.)ANDG:GOSUB2510:F
ORX=.TO2000:NEXT:GOSUB2880:A=A+U         LM
·2460 NEXT:IFA=.THENC$(U)="NO TRAPS AHEAD
.":N=U:BO$="[GREEN]":MC$="[WHITE]":GOTO2
540                                      BF
·2470 C$(U)="YOU AVOID TRAPS.":N=U:BO$="[
RED]":MC$="[WHITE]":GOTO2540             IJ
·2480 A=.:FORD=UTOF:ONDGOSUB2630,2650,267
0,2690:IFNTHENA=A+U:GOSUB2740            LC
·2490 NEXT:IFATHENGOSUB2550:RETURN       BC
·2500 C$(U)="YOU ARE CARRYING NOTHING.":B
O$="[c 7]":MC$="[WHITE]":N=U:GOTO2540    LF
·2510 C$(U)=M$(36+(RND(U)*6))+"!!":N=U:BO
$="[c 3]":MC$="[YELLOW]":GOSUB2560:RETUR
N                                        CN
·2520 C$(1)="YEA":C$(2)="NAY":N=2:AA=18:G
OSUB2570:GOSUB2790:RETURN                BB
·2530 C$(U)="YOU HAVE NOTHING TO "+CM$(I)
+"!":BO$="[PURPLE]":MC$="[WHITE]":N=U    KC
·2540 GOSUB2560:GOSUB2550:RETURN         LI
```

```
·2550 JY=PEEK(JOY):J=FNJ(JY):ON-(FNJF(.))
GOTO2550:WAITJOY,16,16:RETURN                BJ
·2560 AA=11-(N/T)                            PA
·2570 LM=LEN(C$(U))+F:SM=20-(LM/T):GOSUB2
740:RETURN                                   PE
·2580 N=5:FORX=UTON:C$(X)=M$(X):C(X)=X:NE
XT:AA=U:SM=11:LM=18:BO$="[c 5]":MC$="[c
5]"                                          HG
·2590 GOSUB2740:GOSUB2790:IFI=5THENX=-U:R
ETURN                                        OK
·2600 ONIGOSUB2630,2650,2670,2690            EB
·2610 IFN=.THENC$(U)="YOU HAVE NO "+C$(I)
+".":N=U:GOSUB2540:X=-U:RETURN               HG
·2620 X=U:RETURN                             GJ
·2630 N=.:FORX=UTOK+U:IFI(X)THENN=N+U:C$(
N)=I$(X):C(N)=X                              JK
·2640 NEXT:AA=U:SM=T:LM=18:BO$="[c 2]":MC
$="[c 3]":RETURN                             JC
·2650 N=.:FORX=10TO14:IFI(X)THENN=N+U:C$(
N)=I$(X):C(N)=X                              MB
·2660 NEXT:AA=14:SM=U:LM=18:BO$="[c 6]":M
C$="[GREEN]":RETURN                          CD
·2670 N=.:FORX=15TO24:IFI(X)THENN=N+U:C$(
N)=I$(X):C(N)=X                              CN
·2680 NEXT:AA=U:SM=21:LM=18:BO$="[c 4]":M
C$="[c 5]":RETURN                            EP
·2690 N=.:FORX=25TO30:IFI(X)THENN=N+U:C$(
N)=I$(X):C(N)=X                              BL
·2700 NEXT:AA=14:SM=22:LM=17:BO$="[YELLOW
]":MC$="[PURPLE]":RETURN                     JJ
·2710 N=14:FORX=UTON:C$(X)=CM$(X):C(X)=X:
NEXT:AA=T:SM=12:LM=16                        AP
·2720 BO$="[BLUE]":MC$="[c 7]":RETURN        LG
·2730 SYS901,CEM*256,CEM*256+SZ,SS*B(K):S
YS901,CC,CC+SZ,SC*B(K):RETURN                AB
·2740 PRINTCHR$(19);BO$;                     BH
·2750 FORI=UTOAA:PRINT:NEXT:PRINTTAB(SM)L
EFT$(MB$,LM):FORI=UTON+T                     AM
·2760 PRINTTAB(SM)"*"LEFT$(B$,LM-T)"*":NE
XT:PRINTTAB(SM)LEFT$(MB$,LM)                 OM
·2770 PRINTCHR$(19);MC$"[DOWN]":FORI=UTOA
A:PRINT:NEXT                                 IF
·2780 FORI=UTON:PRINTTAB(SM+T);"[RVSOFF]"
;C$(I)::NEXT:RETURN                          AH
·2790 JY=PEEK(JOY):J=FNJ(JY):ON-(FNJF(.))
GOTO2790:PRINTCHR$(19)                       JE
·2800 FORI=UTOAA:PRINT:NEXT:PRINT:I=U        IF
·2810 PRINTTAB(SM+T);"[WHITE]";C$(I)         KF
·2820 JY=PEEK(JOY):J=FNJ(JY):IFFNJF(.)THE
NGOSUB3080:RETURN                            LK
·2830 IFJ<>UANDJ<>TTHEN2820                  PC
·2840 PRINT"[UP]";TAB(SM+T);MC$;C$(I)        EH
·2850 IF(J=T)ANDI<NTHENI=I+U:GOTO2810        NK
·2860 IFJ=UANDI>UTHENPRINT"[UP][UP]";:I=I
-U:GOTO2810                                  PF
·2870 PRINT"[UP]";:GOTO2810                  GL
·2880 SYS901,SS*B(K),SS*B(K)+SZ,CEM*256:S
YS901,SC*B(K),SC*B(K)+SZ,CC:RETURN           PK
·2890 POKEV+34,LV(DL,.):POKEV+35,LV(DL,1)
:POKEV+36,LV(DL,2):Z$=CZ$(DL):RETURN         FG
·2900 POKE56333,127:POKE1,51:SYS901,53248
,55296,61440:SYS901,53248,55296,63488        LK
·2910 POKE1,55:POKE56333,129                 FN
·2920 GOSUB2960:FORX=.TO7:READA:POKE61440
+87*8+X,A:NEXT                               NF
·2930 POKE56578,PEEK(56578)OR3:POKE56576,
PEEK(56576)AND252                            CK
·2940 POKE53272,(PEEK(53272)AND15):POKE64
8,196:PRINTCHR$(147):POKE648,192             KG
·2950 POKE53272,(PEEK(53272)AND240)OR12:R
ETURN                                        NG
·2960 READA:IFA=-1THENRETURN                 DM
·2970 FORX=.TO7:READD:POKE63488+A*8+X,D:N
EXT:GOTO2960                                 KE
·2980 PRINT"[CLEAR][RVSOFF]  [WHITE]A H O
 Y !  M A G A Z I N E  [c 4]PRESENTS[6"[
DOWN]"]"                                     PO
·2990 PRINTTAB(7)"[BLUE]T O M B[3" "]O F[
3" "]H O R R O R[8"[DOWN]"]"                 CH
·3000 PRINTTAB(5)"[c 2][DOWN]THE ULTIMATE
 [WHITE]3-D [c 2]ADVENTURE GAME![DOWN]"      MM
·3010 PRINTTAB(11)"[c 5]JOYSTICK [c 2]IN
PORT TWO[DOWN]"                              PC
·3020 PRINTTAB(6)"(C)1988 [c 5]CLEVELAND
M. BLAKEMORE[DOWN]":RETURN                   PF
·3030 GOSUB3130:POKES+5.K:POKES+6,255:POK
ES+F,23:A=K:D=F:FORX=UTO30:POKES+U,A         DC
·3040 POKES+G,D:D=D*1.01:A=A+U:NEXT:POKES
+6,G:GOSUB3050:GOSUB350:RETURN               DD
·3050 GOSUB3130:POKES+12,F:POKES+19,F:POK
ES+13,241                                    MA
·3060 POKES+20,241:POKES+11,17:POKES+18,1
7:GOSUB50:RETURN                             FL
·3070 FORX=ATODSTEPJ:SYSSR,.,240,X,.,17,1
5:NEXT:GOSUB3050:RETURN                      JG
·3080 SYSSR,.,240,34,75,17,5:GOSUB3050:RE
TURN                                         GB
·3090 A$="2513314215024215O2":FORX=UTOLEN
(A$)STEPH                                    HB
·3100 SYSSR,.,240,VAL(MID$(A$,X,T)),.,33,
10*VAL(MID$(A$,X+T,U)):NEXT                  CO
·3110 GOSUB3050:RETURN                       EG
·3120 SYSSR,.,240,F,F,129,F:GOSUB3050:RET
URN                                          HB
·3130 POKES+24,.:FORRV=.TO24:POKES+RV,.:P
OKES+24,G:NEXT:RETURN                        DP
·3140 DATA32,253,174,32,138,173,32,247       AI
·3150 DATA183,132,193,133,194,32,253,174     JF
·3160 DATA32,138,173,32,247,183,132,195      PC
·3170 DATA133,196,32,253,174,32,138,173      DB
·3180 DATA32,247,183,160,,177,193,145        PP
·3190 DATA20,165,194,197,196,208,6,165       FA
·3200 DATA193,197,195,240,15,230,20,208      LF
·3210 DATA2,230,21,230,193,208,2,230         ON
·3220 DATA194,152,240,225,96,234             IL
·3230 DATA162,25,181,217,9,128,149,217,20
```

```
2,16,247,96                                    IN
•3240 DATA32,230,2,169,15,141,24,212,32,2
41,183,142,5,212,32,241,183,142                FD
•3250 DATA6,212,32,241,183,142,1,212,32,2
41,183,142,,212,32,241,183,142                 OE
•3260 DATA4,212,32,241,183,134,2,162,10,1
60,255,136,208,253,202,208,248,198             GL
•3270 DATA2,165,2,208,240,32,230,2,96,160
,24,169,,153,24,212,136,208,250,96,0           OM
•3280 DATA0,,,,248,240,232,216,184             FK
•3290 DATA28,255,165,165,165,255,165,165,
255                                            OM
•3300 DATA30,31,31,31,31,31,31,31,31           FD
•3310 DATA31,248,248,248,248,248,248,248,
248                                            OM
•3320 DATA38,,,,31,15,23,27,29                 EH
•3330 DATA40,,128,192,224,240,248,252,254      CA
•3340 DATA41,,1,3,7,15,31,63,127               DC
•3350 DATA42,255,,255,255,255,255,,255         BG
•3360 DATA43,128,64,32,16,8,4,2,1              CI
•3370 DATA45,,,,255,255,255,255,255            HF
•3380 DATA47,1,2,4,8,16,32,64,128              NE
•3390 DATA60,255,254,252,248,240,224,192,
128                                            BE
•3400 DATA61,,,,,,,,255                        CJ
•3410 DATA62,255,127,63,31,15,7,3,1,-1         HA
•3420 DATA0,60,126,126,78,78,60,.              GB
•3430 DATA .,,,,,,,,,,,,,,,,.                  PA
•3440 DATA .,4,1,1,1,1,1,1,,1,1,,1,1,1,.       ML
•3450 DATA .,1,,1,,1,,1,1,1,,,,,1,.            PN
•3460 DATA .,1,,1,1,,1,,,1,1,,1,1,1,.          KK
•3470 DATA .,1,1,1,,1,1,1,,1,,,1,,1,.          IG
•3480 DATA .,,,1,,1,,1,,1,1,1,1,1,1,.          KM
•3490 DATA .,1,1,1,1,1,,1,2,,1,,1,,1,.         OO
•3500 DATA .,,,,,,,,,,,,,,,,.                  DO
•3510 DATA .,,,,,,,,,,,,,,,,.                  GA
•3520 DATA .,2,1,1,,1,1,1,1,1,,1,1,1,1,.       PJ
•3530 DATA .,,,1,,1,,1,,1,1,1,,,1,.            GB
•3540 DATA .,1,1,1,,1,1,1,1,1,,1,1,1,1,.       JE
•3550 DATA .,,,1,,1,,1,,1,,1,,1,,.             JF
•3560 DATA .,1,1,1,1,1,,1,,1,1,1,,1,1,.        PD
•3570 DATA .,1,,,1,,1,1,1,1,,1,,,2,.           FJ
•3580 DATA .,,,,,,,,,,,,,,,,.                  KO
•3590 DATA .,,,,,,,,,,,,,,,,.                  JN
•3600 DATA .,4,1,8,1,1,1,,1,,1,1,8,1,1,.       KM
•3610 DATA .,1,,1,,,1,,1,,1,,8,,1,.            PL
•3620 DATA .,1,,1,,,1,1,1,,8,,1,,1,.           MO
•3630 DATA .,8,1,1,8,1,1,,1,1,1,,1,1,1,.       DB
•3640 DATA .,1,,1,,,1,1,,1,,,1,,1,.            IA
•3650 DATA .,1,,8,1,1,1,,,1,1,1,1,,4,.         AC
•3660 DATA .,,,,,,,,,,,,,,,,.                  NL
•3670 DATA160,194,160,152,137,110,185,110
,170,170,170,129,172,150,172,160               HI
•3680 DATASTIRGES,GIANT SPIDER,CONSTRICTO
R,GRAVE RAT,DROW ELF,HOBGOBLIN,FUNGUS          OP
•3690 DATALICH PALADIN,SPECTER,REVENANT,W
YVERN,GUARDIAN NAGA,DEMON KING                 IJ
•3700 DATA 1,1,2,2,3,3,4,4,5,6,6,7,7,8,8,
9,10,10,10,10,11,11,11,12,13                   KH
•3710 DATA 0,,5,9,11,2,5,,13,11,1,5,0          PF
•3720 DATA 18,18,17,2,15,21,19,18,16,3,6,
11,18,4,5,22,8,12,7,13,14,20,18,9,23           MC
•3730 DATA "()","()","'","'","+,","+,","*
","*","./","./","-","-"                        FL
•3740 DATA "12","12","0","0","4687","45",
"3","3","<=>?",":;","9","9"                    OE
•3750 DATA DAGGER,HAND AXE,LANCE,MORNING
STAR,SCIMITAR,BROADSWORD                       LK
•3760 DATA WAR HAMMER,DRAGONSLAYER,EXCALI
BUR,BUCKLER,LEATHER JERKIN,CHAINMAIL           DD
•3770 DATAIRON PLATE,MITHRIL PLATE,COMPAS
S,SCROLL,STEEL HOOK,PINTS OF ELIXIR            AJ
•3780 DATA COIL OF ROPE,BLACK CLOAK,MITHR
IL RING,HOLY SYMBOL,LUMP OF COAL               MC
•3790 DATA LUMP OF IRON,ALCHEMY STONE,GEM
S,JEWELS,GOLD,SILVER,RARE SPICES               JJ
•3800 DATA TAKE,DROP,LEFT HAND,RIGHT HAND
,ARMOR,ASCEND,DESCEND,EXAMINE,INCANT           PB
•3810 DATA QUAFF,HAIL,SEARCH,INVENTORY,EX
IT MENU                                        ON
•3820 DATA 15,12,11,3,14,6,10,8,9              HN
•3830 DATA WEAPON,ARMOR,SPECIFIC ITEM,TRE
ASURE,EXIT MENU,KNOCK YOU SILLY                BH
•3840 DATA BASH YOUR BRAINS IN,GRIND YOU
TO DOGMEAT,TEAR YOU TO RIBBONS                 KC
•3850 DATA GROWLS,BELLOWS,BARKS,ROARS,"PA
SS BY, KNIGHT ERRANT"                          NM
•3860 DATA A SPRAY OF WHITE LIGHT,A CHORU
S OF ANGELIC VOICES                            OK
•3870 DATA IT TRANSFORMS IN YOUR HAND!,A
BLAST OF SPARKS..CHECK INVENTORY               AN
•3880 DATA COMMON,GOOD,EXCELLENT,PRECIOUS
,GLITTERING,FABULOUS,CACHE,CHEST,CRATE         KO
•3890 DATA NO SHAFT LEADING DOWN HERE,NO
SHAFT LEADING UP HERE                          JG
•3900 DATA"YOU DON'T HAVE THE ",THE CATAC
OMBS,THE SEPULCHRE                             LD
•3910 DATA THE BARROW,YOU'RE OUT OF HEALI
NG ELIXIR,YOUR VOICE ECHOES EMPTILY            CP
•3920 DATA POISON DARTS,TRIPWIRE,BEAR TRA
P,SPIKED PITFALL,TRAPDOOR,SNARE NOOSE          HK
•3930 DATA YOU DO NOT HAVE THE ALCHEMY ST
ONE!,T H O U   A R T   S L A I N !             HF
•3940 DATATHOU HAST CONQUERED THE TOMB!        BI
```

TOMB.OBJ

Starting address in hex: C800
Ending address in hex: CFFF

***Flankspeed* required for entry! See page 63.**

```
C800: 08 40 00 1C E0 00 23 10 78
C808: 00 04 82 10 00 07 38 00 DD
C810: 08 C4 00 01 20 00 00 00 FD
C818: 00 00 00 01 08 00 03 9C C0
C820: 00 04 62 00 00 90 00 00 17
C828: 00 00 00 00 84 00 01 CE 7C
```

```
C830: 21 02 31 73 80 48 8C 40 8D
C838: 00 12 00 00 00 00 00 4A
C840: 00 00 00 00 00 00 00 00 40
C848: 00 00 00 00 00 00 00 00 48
C850: 00 00 00 00 00 02 00 40 92
C858: 25 3C A4 54 FF 2A 8D FF 6A
C860: B1 97 FF E1 9B FF D1 FF F8
C868: FF FF DB DD D3 D8 FF 93 62
C870: 48 C1 92 48 41 12 44 22 0F
C878: 22 40 00 02 20 00 04 00 01
C880: 00 00 00 00 00 00 00 00 80
C888: 00 00 F8 00 01 FC 00 02 81
C890: AA 00 02 72 00 02 02 00 B3
C898: 01 04 00 00 88 00 00 70 96
C8A0: 20 00 70 40 00 70 40 00 22
C8A8: 70 60 00 78 30 00 3C 1C 7A
C8B0: 00 1E 0E 00 3C 3C 00 78 CD
C8B8: F0 00 3F E0 00 1F C0 00 A9
C8C0: 00 00 00 38 FF 00 43 FF 3C
C8C8: C0 37 FF E0 0F FF F0 0F B0
C8D0: FF F0 0F FF F0 0E FF B0 80
C8D8: 0E 7F 30 1E BE B8 3E DD 48
C8E0: BC 7F 6B 7E FC FF 9F 78 1C
C8E8: DD 8F 78 49 07 38 7F 03 D9
C8F0: 18 5D 03 0C 08 1E 1E 00 B9
C8F8: 2F 35 00 0A 55 00 14 00 D0
C900: 04 00 40 0C 7C 60 12 BA F9
C908: 90 11 93 10 09 FF 20 07 7D
C910: BB C0 01 C7 00 1F 7D E0 D2
C918: 3F 83 F0 F3 FF 7C A1 7A 58
C920: 1E C0 FC 05 C3 FE 03 C7 8E
C928: CF 03 CE 03 83 CE 01 C6 E6
C930: F7 01 EE E3 87 FC D1 E6 39
C938: 5C 68 67 6A 13 E0 10 04 D6
C940: 34 42 0F 18 66 1F 2C 5A E9
C948: 21 06 66 01 03 5A 01 07 3C
C950: BD E3 0B DB D3 15 E7 AF 59
C958: 2E E7 77 59 7E BD 71 DB C8
C960: 99 60 A5 01 71 99 81 6B F8
C968: 7E C1 66 C3 61 25 81 A1 7C
C970: 0F 00 F1 0F 00 F1 0A 00 7C
C978: 51 0E 00 71 3A 00 5C DC BC
C980: 00 78 0C 70 FC 12 89 FE 0D
C988: 31 C1 FF 64 63 F1 C2 32 2A
C990: 64 C6 64 2A 6C CC AA 66 94
C998: DC 24 E3 7E 71 E3 3F FF 90
C9A0: E6 3E BA FC 1C 82 7A 58 EE
C9A8: 00 31 8C 44 79 87 45 FB EC
C9B0: CF FF FE 7F FF FE 3F FF 3D
C9B8: FE 7F FF FF FF FF FF 04 3B
C9C0: 00 10 00 00 54 00 00 7C A1
C9C8: 00 00 54 00 00 6C 00 00 89
C9D0: 6C 00 01 7D 00 03 AB 80 EA
C9D8: 07 D7 C0 0F EF E0 0D FF 65
C9E0: 60 0C FE 30 0E D6 78 0C E5
C9E8: 6C 10 06 D6 10 00 FE 10 61
C9F0: 00 FE 10 00 FE 10 00 FE 0E
C9F8: 10 01 FF 00 07 FF C0 06 D7
CA00: F8 44 1E CB BB 8E 63 FF D4
CA08: 9F 3B FF 86 03 FF 86 03 F5
CA10: FF 86 03 FF 86 03 FF 86 A9
CA18: 03 FF 86 03 FF 86 03 FF 2E
CA20: 86 03 FF 86 03 FF 82 03 B8
CA28: FF 80 03 FF 80 07 FF C0 F3
CA30: 07 FF C0 0F FF E0 1F FF 07
CA38: F0 7D FF 78 F0 E6 1F FF 16
CA40: 00 10 00 00 10 00 00 92 F2
CA48: 00 00 BA 00 00 BA 00 00 BD
CA50: FE 00 00 92 00 00 D6 00 B8
CA58: 00 EE 00 01 EF 00 0E EE 35
CA60: C0 1F 7D F0 3F AB F8 3F D1
CA68: D7 FC 7F D7 FC 7F EF FC FD
CA70: F7 FF DC F3 FF 9E F3 FF CB
CA78: 9E E3 BB 8E E3 C7 8E 00 7F
CA80: 00 78 00 00 84 00 01 4A C8
CA88: 10 01 02 28 03 87 28 05 7B
CA90: 86 90 0E CD 90 1F 7B D0 7F
CA98: 3D B7 F8 3D CF E4 29 B7 59
CAA0: 64 29 CF 2C 29 FF 38 1A A5
CAA8: 01 18 03 CF 00 03 B7 00 4F
CAB0: 07 CF 00 05 7A 80 0B EF 82
CAB8: 80 0D 7B 40 0D 56 C0 DC 03
CAC0: 03 E7 C0 07 DB E0 07 BD F4
CAC8: E0 0F DB E0 0F E7 F0 0F 6C
CAD0: FF F0 0F FF F0 0F FF F0 C1
CAD8: 0F FF F0 0F FF F0 0F FF E7
CAE0: F0 0F FF F0 1F FF F8 1F 09
CAE8: FF F8 1F 7F D8 3D EF 7C 03
CAF0: 2F BF EC 3A FA AC 2D 95 71
CAF8: F6 5F 56 AE A4 DD CA 66 08
CB00: 00 F8 00 01 04 00 02 02 02
CB08: 00 02 8A 00 02 02 00 02 9A
CB10: 02 00 02 03 06 03 07 09 30
CB18: 03 8F 09 03 DF 09 0D FE AB
CB20: C6 1E FD E6 3F 7B F6 7F 1B
CB28: B7 FF BF CF F7 DF B7 FA F9
CB30: 9F CF F2 9F FF F2 97 FF BC
CB38: D2 67 FF CC 02 00 40 E7 69
CB40: 00 3C 00 00 7E 02 00 42 3F
CB48: 02 00 66 02 00 66 02 3D 58
CB50: BD 82 67 BD C2 77 FF E2 D2
CB58: 43 DB E2 5A FF 77 7E 66 11
CB60: 3F 7E 5A 1E 3C FF 00 01 D3
CB68: 7E 80 03 FF C0 07 81 E0 94
CB70: 0F 00 F0 0E 00 70 0E 00 FC
CB78: 70 0E 00 70 3E 00 7C 1E 40
CB80: F9 BD 8F 70 FF 06 01 FF 3F
CB88: 80 01 FF 80 01 FF 80 03 0F
CB90: 7E C0 06 FF 60 0F FF F0 36
CB98: 1F FF F8 1F 81 F8 1F 00 69
CBA0: F8 1E 00 78 1E 00 78 1E E4
CBA8: 00 78 1E 00 78 1E 00 78 4E
CBB0: 1E 00 78 1E 00 78 1E 00 FB
CBB8: 78 3E 00 7C FE 00 7F FE 69
CBC0: 00 00 00 00 00 00 00 00 C0
CBC8: 00 00 7E 00 00 FF 00 00 47
```

```
CBD0: 81 00 00 81 02 00 E7 02 BF
CBD8: 00 E7 02 00 E7 02 73 7E 9E
CBE0: C2 FF 7E E2 8D 7E E2 CF C3
CBE8: 7E F2 85 FF F2 B7 BD FA 43
CBF0: B5 BD FA B7 FF FA FD FF 10
CBF8: FA FD E7 BA FD 81 BF F8 CC
CC00: 10 B0 10 39 6E 38 38 F9 E2
CC08: 38 38 F0 78 3C F8 78 36 C5
CC10: 76 58 3B 38 B8 3D FF 78 C0
CC18: 3D 7F B8 3D B7 18 3A CB A0
CC20: 38 19 77 D8 08 8F 10 00 69
CC28: FF 00 82 BD 40 47 C3 C0 74
CC30: 4F BD C0 9B C3 C0 B3 3C 0E
CC38: C0 67 81 E0 0A 81 50 00 9E
CC40: 77 FF F7 33 FF E6 12 FF DB
CC48: AC 01 7F 40 01 7F 40 01 77
CC50: 36 40 01 41 40 03 77 60 24
CC58: 03 7F 60 47 36 70 2B 41 95
CC60: 6C 4D 77 5E 9D 7F 7E 9D 29
CC68: 36 5E 9D 41 5C AD 77 5C B9
CC70: AD 7F 6C AD 80 DC F6 FF 0C
CC78: 90 CF 00 3C 14 80 4A 00 F3
CC80: 03 7C 00 09 E7 E0 27 FF F8
CC88: C4 73 F9 4E F3 D8 1F F3 E8
CC90: DC 1F F9 ED 1F FC F7 9F 28
CC98: FE 70 37 F7 78 6F FB BE D9
CCA0: DF FD FF BF F6 FF 7F EF A4
CCA8: FF DF EE FB E3 EE F7 6D AB
CCB0: EE 70 65 ED 86 A9 EF FC 80
CCB8: 6D E7 81 EF F8 3F EF F0 98
CCC0: 00 FC 00 01 7A 00 03 B7 F3
CCC8: 00 03 CF 00 01 FE 00 00 9B
CCD0: FC 00 00 78 00 00 78 00 BE
CCD8: 00 78 00 00 78 00 00 78 42
CCE0: 00 00 78 00 00 78 40 00 12
CCE8: 78 20 00 78 20 00 FC 60 77
CCF0: 01 FF C0 0F 00 E0 1F FF C1
CCF8: F0 18 03 F0 0F FF E0 DD C3
CD00: 03 BD C0 03 DB C0 02 BD E0
CD08: 42 02 DB 44 03 BD C6 03 F6
CD10: DB C3 03 BD C3 02 DB C7 D9
CD18: 05 E7 CF 0B FF CE 17 FF C5
CD20: DC 2F FF DC 5F FF DC 5F A4
CD28: FF BE 7F FF 7E 7F FE FE 62
CD30: 7F FE FE 7F FE FE 3F FF 6A
CD38: 7C 1F FF F8 07 FF F0 1C E0
CD40: 07 E7 E0 0B 7E D0 15 3C BB
CD48: A8 2E FF 74 5F 7E FA 5F CB
CD50: 81 FA 5F BD FA 2F FF F4 09
CD58: 14 FF 28 0B 00 D0 04 FF 74
CD60: 20 03 81 C0 02 E7 40 03 F2
CD68: 99 C0 03 DB C0 02 BD 40 62
CD70: 02 DB 40 03 BD C0 02 DB ED
CD78: 40 02 BD 40 03 DB C0 00 58
CD80: 03 FF C0 03 FF C0 03 DB E6
CD88: C0 03 BD C0 02 DB 40 02 EA
CD90: BD 40 03 DB C0 03 BD C0 AF
CD98: 03 DB C0 03 DB C0 02 BD 97
CDA0: 40 02 DB 40 03 BD C0 03 83
CDA8: DB C0 02 BD 40 03 DB C0 E4
CDB0: 02 BD 40 02 DB 40 03 E7 B9
CDB8: C0 03 FF C0 03 FF C0 01 02
CDC0: 7F 00 00 FE 80 00 FB 40 FB
CDC8: 00 F3 20 00 E6 D0 00 FD 92
CDD0: A8 00 EB D4 00 FB EA 00 21
CDD8: 07 EA 00 CF EA 00 DF DA 40
CDE0: 00 FF B4 00 FF 68 00 00 FD
CDE8: D0 00 FD A0 00 03 40 00 9B
CDF0: FC 80 00 FB 00 00 FA 00 65
CDF8: 00 D4 00 00 D4 00 00 DC 7F
CE00: 00 00 FE 00 01 7F 00 02 81
CE08: DF 00 04 CF 00 0B 67 00 2E
CE10: 15 BF 00 2B D7 00 57 DF 1F
CE18: 00 57 E0 00 57 F3 00 5B F6
CE20: FB 00 2D FF 00 16 FF 00 5F
CE28: 0B 00 00 05 BF 00 02 C0 BA
CE30: 00 01 3F 00 00 DF 00 00 50
CE38: 5F 00 00 2B 00 00 2B DC CA
CE40: 1C 00 70 3E 00 F8 3F 6D B0
CE48: F8 3F C7 F8 3F FF F8 3B B4
CE50: BB B8 31 93 18 23 6D C8 FA
CE58: 07 A9 E0 0D BA E0 0C C2 61
CE60: 60 0E 7D C0 07 FA 80 01 90
CE68: BA 80 00 FD 40 00 EE 00 D0
CE70: 01 C7 00 01 C3 80 00 C3 42
CE78: 80 00 C3 00 03 42 C0 00 C2
CE80: E3 FF DE E1 F5 C6 C1 FA 9E
CE88: FF BA F5 C7 7F 75 41 05 3C
CE90: BA C0 04 BF C0 09 7F F0 0A
CE98: 00 F0 F0 01 E0 70 03 C0 90
CEA0: F0 03 C0 F0 03 C0 E0 01 EB
CEA8: C1 C0 01 C3 80 01 C3 80 B5
CEB0: 01 C3 80 03 C1 C0 03 C1 40
CEB8: E0 07 C1 F0 1D 22 5C 00 EE
CEC0: 60 00 1C F0 00 1E F8 00 45
CEC8: 3F F8 00 3F FC C6 7F EE 72
CED0: 83 7F ED 83 AF ED C3 AF 56
CED8: ED FF AF EF 7E EF F7 3C 09
CEE0: DF 73 99 8E 22 FF 44 0C CE
CEE8: EB 30 1F 6B 78 7F BE F8 3F
CEF0: 77 DD F8 E5 E3 B8 C2 FF 84
CEF8: 5C C3 18 CE E1 FF 8E 00 70
CF00: 0F F0 FE 0F E0 FE 0F C0 BD
CF08: 7E 0F C0 7E 1F 80 3E 1F D1
CF10: 00 3E 1F 00 7E 1F 00 FC 08
CF18: 3F 00 F8 3F 01 F8 3F 01 C9
CF20: F0 1F 01 F0 1F 03 E0 0F 34
CF28: 03 E0 0F 47 C0 07 87 C0 72
CF30: 03 87 C0 03 87 80 07 87 15
CF38: C0 0A C7 60 14 8A 90 FC 57
CF40: 7F BF 7E FB C0 EF FB F7 9E
CF48: EF FB F7 EF FD FF EF F6 01
CF50: 0C 1F F7 FF EF F3 FC 6F C3
CF58: F0 7B AF E3 B7 E7 EF D7 BF
CF60: 37 FF B5 3F FC B4 9F 79 57
CF68: 5A EF 31 39 E0 01 FF E0 DF
```

```
CF70: 03 FF E0 03 FF F0 03 FF 4B
CF78: F8 07 E1 FC 0F E0 FE FC 43
CF80: 07 80 00 0F E0 00 1F F0 08
CF88: 30 3F F0 E0 7F F9 C0 FF 04
CF90: FD C0 FF FD C0 FF DD EF DB
CF98: FF DD FF FF DE FF FF BE 14
CFA0: BF FF BD 9F FF BB 8F FF 09
CFA8: BB C3 FF CB FF FF 93 FF 87
CFB0: FF 1B FF FE 15 FF 7C 3E 9A
CFB8: 3D 38 3F DD 10 1F EF FC 67
CFC0: 00 01 E0 00 07 F0 0C 0F B5
CFC8: F8 07 0F FC 03 9F FE 03 79
CFD0: BF FF 03 BF FF F7 BB FF 07
CFD8: FF BB FF FF 7B FF FD 7D 8B
CFE0: FF F9 BD FF F1 DD FF C3 2C
CFE8: DD FF FF D3 FF FF C9 FF 64
CFF0: FF D8 FF FF A8 7F BC 7C 2B
CFF8: 3E BB FC 1C F7 F8 08 FC 02
```

SCUTTLEBUTT

Continued from page 12

plus free shipping.

P.A.V.Y. Software, 314-527-4505 (see address list, page 12).

AI TYPING TUTOR

Mavis Beacon Teaches Typing by use of artificial intelligence, checking the student's progress every step of the way and tailoring an individualized course of study. Mistakes are explained in conversational sentences, rather than by bleeps and bells. Customized practice lessons include literary quotes, riddles, and facts from the Guinness Book of World Records. Other features are a Road Racer typing game, and musical "tempo typing" with onscreen metronome. Over 20 different graphs display the student's progress. $39.95 for the C-64; $44.95 for the Amiga.



A practice screen from **Mavis Beacon.**
READER SERVICE NO. 178

The Software Toolworks, 818-907-6789 (see address list, page 12).

UNSILENCED

MicroProse's *Silent Service* is again on sale in West Germany after a government agency in Bonn rescinded a 14-month old prohibition. The WWII submarine simulation had been judged potentially harmful to youth. But MicroProse called into question the agency's criteria for determining the product harmful, and contended that banning sales to youth effectively banned sales to adults too, since the product could under those circumstances not be advertised, or sold except in stores with adults-only entrances. The ban was lifted less than 24 hours before a scheduled hearing.

With the matter settled, there's little need for editorializing—but we applaud the West German government for overturning its earlier decision. The events of 50 years ago are understandably not a source of national pride for modern-day Germans. But there's more to be gained by studying and understanding the past than by attempting to strike it from the record.

MicroProse, 301-771-1151 (see address list, page 12).

REVIEWS

Continued from page 56

tem that would not require a customer to wait while you swapped disks and entered most information twice.

With those caveats, we'll go on to say that the documentation, though slight, is adequate, and that most people should find the system easy to use. Much of this is due to onscreen prompts and messages that will lead you through each step of the transaction, many of them highlighted by flashing graphics that are designed to recapture wandering attention spans.

System requirements for using *TI-SAR 128*—with no deviations tolerated—are a C-128 or C-128D, a 1571 drive, and an 80-column monitor with color preferred. All disks in the system are copyable so that you can keep originals in a safe place, and each program has a "disk-doctor" to help retrieve information lost due to accidents.

CW Data Labs, 1632 Napfle Avenue, Philadelphia, PA 19111. —*Ervin Bobo*

WARPSPEED!
CinemaWare
Commodore 64/128
Cartridge; $49.95

The Commodore 64/128 has many virtues. Processing speed is not among them. Once, a computer magazine even printed a list of things for Commodore users to do while waiting for a program to load from disk.

Several manufacturers have attempted to address this annoying shortcoming with "speed up" cartridges. *Mach-5* and *Mach-128* (Access) and *Fast Load* (Epyx) are probably the best known.

Now Cinemaware has entered the field with what may prove to be the most powerful and versatile accelerator cart on the market. Though the company earned its reputation with entertainment products like *Defender of the Crown* and *King of Chicago*, skeptics will quickly become converts once they try this amazing booster box.

Like all existing accelerator units, *Warpspeed!* plugs into the Commodore's cartridge slot. It has a mode switch to select either 64 or 128 and a recessed reset button.

Warpspeed! conveniently supports the *Fast Load* command structure, including the two-key (RUN STOP and COMMODORE) combination, which eliminates typing information to load a program. It loads, saves, and formats disks up to 10 times faster than normal. A few disks, like the QuantumLink terminal program, won't load with *Warpspeed!*, but most do.

The copy features of this menu-driven cartridge are even more impressive. It duplicates disks and files up to 40 times faster. The user can even pick several files from the same disk and copy or scratch them in one operation.

Warpspeed! supports both double- and single-sided drives, including the 1581 and MSD units. It copies files between single- and double-sided disks and enables the 1571 to function in dou-

ble-sided mode on the C-64.

Cinemaware's cartridge is also packed with goodies for programmers and "serious" users. *Warpspeed!* boasts its own mini-assembler and built-in machine language monitor. These functions, which work in disk drive memory, include the ability to "smart move" code up and down in RAM. The cartridge allows relocated loads and saves, verifies disk files from inside the monitor, can redirect output to the printer, loads/saves disassembles under ROMs and I/O ($d000), and incorporates an extended DOS wedge.

The machine language monitor and mini-assembler interface with the sector editor also contained in *Warpspeed!*. It edits in hex or normal ASCII text and traces directory and file track and sector links.

That may sound like a lot of features in a small package, but these are only the high spots. *Warpspeed!* has over 30 tools for casual and advanced users. Alien Technologies, a British-based development company, has clearly tried to create the ultimate accelerator cartridge. Future products may topple the new king of the speed carts from its throne, but right now, *Warpspeed!* is an indispensible aid for every Commodore computer owner.

Cinemaware, 4165 Thousand Oaks Blvd., Westlake Village, CA 91362 (phone: 805-495-6515). *–Arnie Katz*

TIPS AHOY!

Continued from page 44

```
•110 POKE R,255-PEEK(R):NEXT
•120 POKE 4588,56:CHAR,0,0,"[c T][c I][c
 P][c S][SS][c A][c H][c O][c Y][cEP]"
•130 POKE 2604,30:WINDOW 0,0,39,24,1
•140 FOR A=160 TO 191:X$=X$+CHR$(A):NEXT
•150 SYS DEC("CC6C"),,18,4:PRINT X$:A=179
 1+RND(0)
•160 CHAR ,4,2,X$:BOX 1,30-B,30-B,289+B,9
 8+B
•170 A=A+1:X=SIN(A/2)*1.9+2:Y=COS(A/[PI])
 *1.9+2
•180 CIRCLE 1,91,164,X,Y,,,,30:CHAR ,0,20
 ,"+"
•190 CHAR ,4+(AAND31),4+(7ANDA/32),"+"
•200 CIRCLE 0,91,164,X,Y,,,,30
•210 GET A$:IF A$="" THEN 170
•220 FORA=14592TO14599:POKEA,0:NEXT
•230 FOR A=0 TO 13:CHAR ,0,A,"[40" "]":NE
 XT
•240 GRAPHIC0:LIST
```

PROGRAM PAUSER

Most of us have heard of and used program pausers. With the press of a key, the list or execution of a program is stopped in its binary tracks, until another key is pressed. *Program Pauser* takes this a step forward.

Load *Program Pauser* and change the variable PSWD$ to a secret code. (The default is "AHOY!") Run the program. Then, when you wish to pause your 128, press CONTROL and the up arrow key. The computer will pause. To restart the computer on its merry way, press RETURN, the secret code, and RETURN.

–Shawn K. Smith
Bronx, NY

```
•100 REM PROGRAM PAUSER -- SHAWN K. SMITH
•110 FORD=6912TO6999:READY:POKED,Y:NEXT
•120 PSWD$="AHOY!":L=LEN(PSWD$)-1
•130 POKE6955,L:POKE6976,L:FORD=DTOD+L
•140 POKED,ASC(MID$(PSWD$,L+1-P,1))
•150 P=P+1:NEXT:END
•200 DATA 169,023,141,060,003,169,027,141
•210 DATA 061,003,096,169,173,162,198,141
•220 DATA 060,003,142,061,003,096,234,201
•230 DATA 094,240,003,076,173,198,032,011
•240 DATA 027,173,004,010,041,254,141,004
•250 DATA 010,234,160,002,132,251,032,228
•260 DATA 255,240,251,201,013,240,243,164
•270 DATA 251,153,093,027,136,016,237,160
•280 DATA 002,185,088,027,217,093,027,208
•290 DATA 225,136,016,245,173,004,010,009
•300 DATA 001,141,004,010,076,000,027,234
```

PRETZEL

Here's a quick and easy demo of the power of C-128 graphics and basic trigonometry. Try changing COLOR2 and COLOR3, increment of "P", draw coords, and trig formulae. Add a "DRAW1..." statement for a third color. Or "stack" drawings by changing "GRAPHIC3,1" to "GRAPHIC3,0" and changing other aspects after your first run. The possibilities are truly endless....

–William D. Wolfe
Pittsburgh, PA

```
10 WIDTH2:COLOR0,1:COLOR4,1:GRAPHIC3,1:D
0:P=P+.01:X=60*COS(P*3)+68:Y=60*COS(P*4)
+105:DRAW3,X+22,YTOX,Y-9:DRAW2TOX,Y:LOOP
```

...COMING IN THE SEPTEMBER ISSUE OF *AHOY!* (ON SALE AUGUST 2)...

COMPILER BASIC
HIGH-SPEED PROGRAMS

GETTING INTO PARSERS
TEXT GAME PROGRAMMING, PART III

FIREBIRD/AHOY! CONTEST RESULTS
16 TOP ENTRIES PUBLISHED